

Mobile Autonomous Router System for Dynamic (Re)formation of Wireless Relay Networks

Kyu-Han Kim, *Member, IEEE* Kang G. Shin, *Fellow, IEEE* Dragoş Niculescu

Abstract—Multi-hop wireless relays can extend the area of network connectivity instantly and efficiently. However, due to the spatial dependency of wireless link-quality, the deployment of relay nodes requires extensive, expensive measurement and management efforts. This paper presents a *Mobile Autonomous Router System*, (MARS) through which a relay router autonomously seeks and adjusts the best “reception” position for itself and cooperatively forms a string-type relay network with other neighboring routers. Specifically, MARS (i) accurately characterizes spatial link-quality through a new measurement technique, (ii) effectively probes/optimizes node positioning via a spatial probing algorithm, and (iii) maintains error-tolerant position information via an inexpensive positioning algorithm. MARS has been prototyped with both a commodity mobile robot and a wireless router with IEEE 802.11 cards. Our experimental evaluation of both the MARS prototype and ns-2-based simulation show that MARS achieves an average of 95% accuracy in link-quality measurements, and reduces the measurement effort necessary for the optimization of a node’s location by two-thirds, compared to exhaustive spatial probing.

Index Terms—Wireless relay networks, robot-based wireless router, wireless link-quality measurement, IEEE 802.11

I. INTRODUCTION

Over the last decade or so, wireless relay networks have attracted considerable attention due to their potential for instantly and inexpensively extending network coverage [1], [2]. For example, relay nodes can be deployed in a string or a tree topology to existing wireless backhaul networks for emergency response or outdoor events. However, channel fading and shadowing often degrade the quality of wireless links and require time-consuming and expensive network deployment and management efforts, especially for manual adjustment of nodes’ placement and configuration [3].

Significant efforts have been made to improve Quality-of-Service (QoS) and reduce management costs of wireless relay networks. For example, measurement-driven deployment of relay nodes determines their placement positions that meet the required network QoS [4]. Rate-adaptation and transmission-power-control algorithms allow for dynamic selection of modulation schemes and transmit-power levels at fixed positions [5], [6]. Multiple-input-multiple-output (MIMO) or multiple interfaces enable a node to exploit spatial diversity by adaptively choosing the best antenna or antenna element [7], [8].

Kyu-Han Kim is currently with Hewlett-Packard Laboratories in Palo Alto, CA 94304, U.S.A. Kang G. Shin is with the Real-Time Computing Laboratory in the Department of Electrical Engineering and Computer Science at The University of Michigan, Ann Arbor, MI 48109, U.S.A. Dragoş Niculescu is with the Telecommunications Department at the University Politehnica of Bucharest Romania. E-mail: kyu-han.kim@hp.com, kgshin@eecs.umich.edu, dragos.niculescu@elcom.pub.ro

In spite of these efforts, wireless relay networks still suffer from several fundamental limitations as follows. First, changes in the physical environment of a wireless relay network (e.g., due to dynamic obstacles and interferences) often calls for *manual* link-quality measurement and node relocation over a large coverage area which are tedious, time-consuming and costly. Second, even if relay nodes can make simple movements to improve link bandwidth, these adjustments determined by geographic distance or node density are not guaranteed to meet the overall QoS requirement of networks [9], [10]. Third, relay nodes may be able to relocate themselves “optimally” (in some sense) by using simple Signal-to-Noise Ratio (SNR) or traffic volume information. However, accurate characterization of spatial wireless link-quality along with efficient node relocation is key to the efficient (re-)formation of relay networks [11].

In this paper, we study the feasibility of using a commodity mobile robot for addressing some of the limitations mentioned above. Specifically, we propose a *mobile autonomous router system* (MARS) that enables a wireless relay node to (i) characterize wireless link conditions over the physical space and (ii) seek and relocate to, the best reception position to form *string-type* relay networks. Specifically, MARS is equipped with a measurement protocol that defines and characterizes spatial wireless link-quality. This measurement protocol controls each MARS node, mounted on a mobile robot, to move and measure wireless link-quality over the deployment area. Furthermore, based on the raw measurement results, the protocol extracts unique correlations of link-quality with environmental factors, such as distance, obstacles, or interference sources, which are useful in reducing the measurement space (see Section V-C). Here, MARS focuses on the string-type relay network that allows for the easy deployment of extended backhaul links between remote end-points. Applications for such a network include home networks (e.g., extension of access point’s range), outdoor events (football stadium) and environmental (e.g., US border) monitoring. However, we will also describe how MARS can support other topologies such as a tree type in Section V-B.

Next, MARS includes a spatial probing algorithm by which the node can efficiently find its optimal position that satisfies the bandwidth demand on its link. Alternating between measurements and movements, this probing algorithm guides the robot to identify an ‘interesting’ space to probe. This space is then explored at progressively finer resolutions until a locally optimal position is found. Moreover, the algorithm enables a set of MARS nodes to *cooperatively form and adjust wireless relay networks* in case link conditions or QoS demands change.

Note that MARS focuses on QoS demands of backhaul links, excluding client-to-access node links. We assume that access node (i.e., MARS) can aggregate QoS demands from multiple clients that are associated with it and then use the aggregate demands to form relay links.

MARS also includes a light-weight positioning system that provides location information to the robot. This system is currently implemented for indoor environments and does not require any expensive infrastructure support, such as cameras and other sensors, but uses natural landmarks, which are easily obtainable with a semi-automated collection procedure, as detailed in Section VI-C. Moreover, even though MARS is designed for challenging indoor environments, such as office buildings or large retailer shops, it is flexible enough to use any type of positioning system, depending on deployment scenarios (e.g., Global Positioning System (GPS) in outdoor environments).

A prototype of MARS has been built with commodity mobile robots and IEEE 802.11-based wireless routers, and its algorithms have been implemented in Linux using a combination of kernel- and user-space functionalities. Our experimental evaluation results on a prototype implementation of MARS indicate that MARS achieves an average of 95% accuracy for link-quality measurements, and finds locally optimal locations with 300% less measurement overhead than exhaustive spatial probing. Furthermore, our simulation results using *ns-2* [12] show that MARS reduces energy usage by up to 54% during connectivity reformation.

In summary, this paper makes the following main contribution:

- We study the feasibility of using a mobile robot for dynamic (re)formation of a string-type wireless relay network, while reducing manual measurement and relocation overheads.
- We design, develop, and implement MARS that autonomously characterizes spatial link-quality and uses/controls mobility to find (sub)optimal position of a wireless router for QoS support.
- We show the effectiveness of MARS through extensive performance evaluation using both system prototype and simulation.

The rest of this paper is organized as follows. Section II provides the motivation behind this work. Section III presents the software architecture and a hardware prototype of MARS. Sections IV–VI detail the core components of MARS. Section VII presents the evaluation results of our MARS implementation. The paper concludes with Section VIII.

II. MOTIVATION

We first argue for the need of a mobile autonomous router system (MARS) in wireless relay networks, and then discuss why existing techniques are not suitable for MARS.

A. Why Mobile Autonomous Routers?

Due to open and continuously-changing deployment environments, wireless relay networks often incur high measurement and (re)configuration costs [13]–[15]. After their

deployment, relay networks may often experience severe QoS degradation, require additional measurements, and/or need to adjust placement of the relay nodes, as their physical environment changes. Even though various techniques (e.g., transmission-power control, rate adaptation) and technologies (MIMO, multi-radios) have been proposed, their bandwidth improvement is essentially limited by the surrounding environment. For example, assuming that nodes already use their maximum transmission power, a *stationary* node behind the wall might not be able to improve the bandwidth of link to another node on the opposite side of the wall using the existing techniques.

In contrast, by utilizing their mobility, mobile wireless relay routers can overcome the spatial dependency of link-quality. Being aware of diverse link-quality at different locations, mobile wireless routers can automatically improve network performance and also offer several benefits in wireless relay networks as follows.

- *Extension of AP's range*: users near the boundary of an AP's coverage might experience intermittent connectivity. A mobile wireless router can be placed near the limit of the AP's communication range and relay users' traffic for the extension of the AP's range.
- *Deployment of wireless relay networks*: for rural areas or outdoor events, multi-hop relay networks are an inexpensive way to provide/extend connectivity, but their optimal placement is still a challenging task. A group of mobile routers can identify a proper position of each router, forming a relay network [16].
- *Adjustment of wireless routers' placement*: many nodes in a relay network need to re-adjust their position to deal with environmental changes (e.g., dynamic obstacles in urban areas), but manually adjusting locations of each pair of nodes incurs significant time and management costs [17]. Equipped with (albeit limited) *local* mobility, mobile routers can cooperatively and automatically adjust their locations to improve connectivity and link-quality.
- *Improved resilience to DoS attacks*: wireless networks are vulnerable to denial-of-service (DoS) attacks such as jamming, and channel hopping provides limited resilience to wide-band jamming. However, because of the spatial locality of such attacks, mobile wireless routers in a relay network can improve resilience to such attacks by physically moving away from the jamming source [11].

Motivated by the above and other likely scenarios, we would like to (i) design an autonomous robot-based router system that accurately captures the quality of wireless link over space and that effectively optimizes a router's position based on the thus-obtained characteristics, and (ii) prototype and evaluate such a mobile router, especially for *string-type* relay networks. Specifically, as illustrated in Figure 1, we focus on the placement of mobile wireless routers from an AP or a gateway and the formation of multi-hop relay links among the routers given links' QoS requirements.

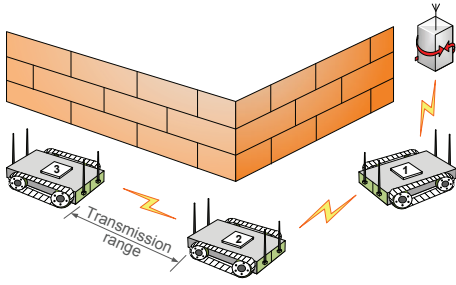


Fig. 1. A group of mobile routers can cooperatively move and form string-type wireless relay networks by being aware of spatially diverse link-conditions over deployment areas.

B. Limitations of Existing Approaches

There has been a significant volume of work on link-quality awareness and the use of mobility. We discuss pros and cons of using existing approaches for the design of a mobile router.

Link-quality awareness: Mobile routers (relay nodes) must be able to accurately measure wireless link-quality—In this paper, we consider the packet-delivery ratio or bandwidth as link-quality parameters of interest—over physical space. There have been numerous link-quality measurement studies and techniques in wireless mesh networks, large-scale WLANs, and wireless sensor networks [18]–[22]. Their insights and solutions, however, focus on stationary networks where the physical space of measurements is fixed. The authors of [23] considered the link-quality change resulting from location changes in mobile ad-hoc networks (MANETs). However, their solution only deals with a binary (ON/OFF) connectivity model based on simple metrics such as SNR and distance, and thus, is not suitable for capturing various channel conditions in diverse deployment areas [9].

Node mobility: Mobile routers must be able to exploit their mobility in conjunction with link-quality-awareness. Mobility in wireless networks has been considered from various perspectives. First, flip-type mobility has been proposed for sensor nodes to make a one-time random movement [24]. Second, use of sophisticated robots in hybrid sensor networks has been proposed, and the tradeoff between node density and node mobility has been studied [25]. Third, using mobile sensors against radio jamming attacks has been considered, and movement-decision metrics (i.e., SNR and the amount of traffic) have been proposed [11]. Fourth, multipath-fading-aware mobility control strategies to improve link throughput has been proposed in [26]. Finally, in MANETs, users’ mobility is exploited for mobile devices to improve their end-to-end throughput [27], [28]. However, the mobility used in these existing approaches is not closely coupled with link-quality-awareness or based on a theoretic channel model, reducing the chance of making optimal movements in heterogeneous network deployment environments.

Position-awareness: Mobile routers with limited battery and CPU power must be equipped with a light-weight positioning system that does not require any extensive computation or expensive infrastructure support. On the one hand, positioning

in an open sky outdoor environment is relatively easy and fast since GPS provides accurate position information. On the other hand, positioning in an indoor environment or an outdoor urban canyon is challenging, and many solutions have been proposed. In the area of robotics, the use of different hardware (sonar, laser, compass, and video camera) has been explored [29], [30], and various assumptions (e.g., known or unknown landmarks) and algorithms (training or search) have also been investigated [31]. However, their hardware and computation costs have to be carefully considered, depending on the underlying applications. In the field of wireless networks, using AP-based landmarks [32], ceiling-mounted sensors and listeners [33], and video cameras [34] have been proposed, but these approaches need installation of a positioning infrastructure or cooperation from the network infrastructure (e.g., APs).

III. MARS ARCHITECTURE

We now present the architecture of MARS. We first describe its design rationale and software architecture. Then, we present our current hardware prototype of MARS that is used for link-quality measurement and system evaluation.

A. Design Rationale

MARS is a distributed system in which each MARS node autonomously adjusts its position to meet the network (e.g., bandwidth and network coverage) requirements via the following distinct features.

- *String-type relay networks:* MARS supports the dynamic (re)formation of a string-topology wireless relay network. Using commodity robots and IEEE 802.11 wireless cards, MARS aims to extend wireless network coverage where each relay link satisfies given network requirements, as illustrated in Figure 1.
- *Patch-based spatial measurement:* To characterize link-quality over a wide area, MARS divides space into fixed-size squares (or *patches*) and measures the link-quality in the selected patches. This divide-and-conquer approach enables MARS to locate subspaces or subareas where the link quality meets the QoS requirements within the entire deployment area.
- *Hierarchical spatial probing:* MARS takes a hierarchical approach to finding locally optimal positions. Unlike exhaustive spatial probing, MARS incrementally probes space as needed. By repeating the cycle of spatial measurement and local spatial-probing decision, MARS reduces the probing overhead.
- *Infrastructure-less hybrid positioning:* MARS includes a positioning system for the case where no external positioning system is available. By exploiting the benefits of both dead-reckoning and physical-landmark-based positioning, MARS achieves positioning accuracy at a reasonable training cost.

Throughout this paper, we assume that a robot has only basic driving capabilities (forward/backward/spin), which are adequate for evaluating the proposed design of mobile routers.

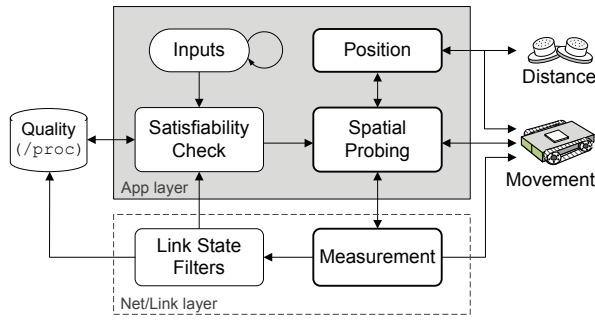


Fig. 2. *MARS Software architecture.* MARS is designed across the application, network, and link layers in a router, and controls sensors for measuring distance and a robot for movement.

Further, we consider link bandwidth (or packet delivery ratio (PDR)) as the QoS parameter, and the two terms are used interchangeably.

B. Software Architecture and Operations

Following the above rationale, the software architecture of MARS is designed as shown in Figure 2, and operates as follows. Initially, a MARS node receives the bandwidth requirement of link to AP or a neighboring node and then checks if the node’s current position meets the requirement, labeled as *Satisfiability Check* in the figure. If the current position does not meet the requirement, MARS then decides which direction it has to move-and-measure (*Spatial Probing*) based on previous link-quality information (*Quality*). Next, if further measurements are necessary, MARS moves to a different location (*Movement*) and measures the link-quality at the new location (*Measurement*). Based on the measured link-quality, MARS again checks the bandwidth satisfiability. This procedure is repeated until MARS finds the position that satisfies the bandwidth requirements. Finally, for each movement, MARS maintains the node’s position information (*Position*) using the distance information periodically measured by sonar sensors (*Distance*).

Next, we will detail the three core components of MARS: spatial measurement protocol in Section IV, spatial probing algorithm in Section V, and positioning system in Section VI.

C. Hardware Prototype

Before detailing the algorithms of MARS, we describe its hardware prototype built for our measurement and evaluation of MARS. Figure 3 depicts the hardware prototype of MARS, which consists of a mobile robot, a multi-radio wireless router, and sonar sensors: (i) MARS uses iRobot Create [35] for mobility, which provides a well-defined API for movement control (e.g., a granularity of 1cm movement) and is powerful enough to carry a network node as in [34], [36]; (ii) MARS is equipped with an RB230 wireless router (233Mhz CPU, 128MB memory) [37], and the router is equipped with two IEEE 802.11 miniPCI NICs, each with a 5 dBi omnidirectional antenna. In addition, this router includes a serial port for communication with the robot and sonar sensors, and is connected to an external battery for a long lifetime;

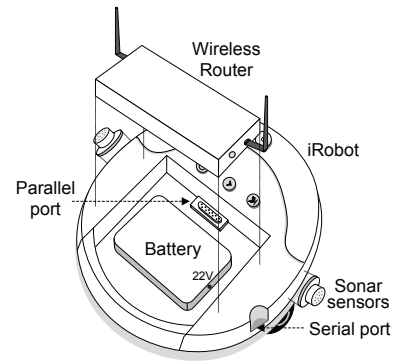


Fig. 3. *MARS hardware prototype.* A MARS node is prototyped with an iRobot, a wireless router, and sonar sensors.

(iii) MARS is equipped with a sonar sensor on each side of the robot to measure the distance between the robot and the surrounding obstacles. The sonar sensor is cheap (about \$25 apiece) and provides accurate distance information (error of less than 2.5cm) to objects placed at up to 6m in line of sight for the positioning system.

IV. MEASUREMENT PROTOCOL

We first overview challenges in designing a link-quality measurement protocol, and then propose a measurement protocol that characterizes spatial link-quality through point and spatial measurement techniques.

A. Overview

Key challenges in measurements are (C1) how to measure the link bandwidth at a given position/time and (C2) how to characterize the quality of links as a function of physical space. First, to determine if the current position’s link satisfies the bandwidth demand, the measurement protocol has to be able to accurately estimate the link bandwidth at each position. Moreover, the protocol has to capture the link bandwidth as quickly as possible, to reduce probing time, or equivalently node’s energy consumption.

Next, MARS must be able to derive the overall quality of links in a certain space, mainly for intelligent selection of a subspace for measurement within a large deployment area. Even though wireless link-quality may vary with the node’s location [38], MARS needs to be able to characterize and differentiate spaces with respect to the quality of links. In addition, this characterization must be accurate even in the presence of adverse environmental factors, such as moving obstacles or short-term interference during measurement period, which may cause temporal variations in link-quality measurement.

The measurement protocol in MARS uses both point and spatial measurement techniques to overcome the challenges mentioned above. In what follows, we will detail both techniques with which MARS derives spatial link-quality together with measurement results. Note that in this section, we use controlled configurations (e.g., low transmission power, small measurement size) for experiments, due to indoor space constraints. Although the absolute numbers of results might

be different under different settings, their trends support the underlying rationale of the proposed techniques. We will also present evaluation results under real-life settings in Section VII.

B. Point Measurement Technique

To meet the accuracy and time constraints, MARS includes a *unicast-based active probing* technique along with a *cross-layer design principle*. There have been numerous techniques for estimating wireless link bandwidth. First, the SNR-based approach has been studied extensively for the PHY layer based on information theory [38], but the correlation between SNR-based bandwidth estimation and packet-level bandwidth estimation is shown to be weak [9], [39]. Second, the broadcast-based packet-delivery ratio (PDR) measurement has been widely used in multi-hop routing metrics, such as ETX (Expected Transmission Count [40]) and ETT (Expected Transmission Time [41]). However, due to different underlying communication settings (e.g., modulation schemes) of the broadcast from those of actual data transmission, the technique has limited measurement accuracy, even if the broadcast is modified to use various modulation schemes [42], [43]. Next, the packet-pair technique used in [41] is bandwidth-efficient, but sensitive to the short-term fading effect due to its use of a small number of probing packets. By contrast, the measurement technique in MARS uses a set of unicast probing packets to mimic actual data transmissions for accuracy, and evenly spaces the packets throughout a given measurement period (e.g., 1 packet every 50ms for a x second-period) to minimize the undesirable effect of channel fading (e.g., bursty bit-errors). Here, the measurement period can be given as a system parameter, and this parameter can be determined, depending on site-specific radio environment as well as system tolerance to temporal measurement overhead.

Furthermore, as shown in Figure 2, the measurement technique in MARS is designed and implemented across the network and link layers to further improve measurement accuracy. After receiving a point measurement request from the spatial probing algorithm, the measurement protocol at the network layer sends a set of unicast probing packets to a neighboring node. At the same time, the protocol at the device driver passively monitors their transmission results based on MAC's feedback. The use of this feedback at the sender side allows for capturing the total number of (re-)transmissions made by MAC for delivering the probing packets, yielding an accurate PDR of the link. Next, to capture link asymmetry, MARS requests the neighboring node to execute the same probing procedure in the opposite direction, and then the node sends the measurement result back to the MARS node. Finally, the PDRs of both directions are stored in the *Quality* database at the MARS node, and the protocol notifies the completion of the point measurement to the spatial probing algorithm.

Our measurement evaluation of the above protocol shows that the active probing technique in MARS achieves higher than 95% accuracy in estimating link bandwidth. Figure 4 shows the progressions of actual UDP throughput (upper figure) and PDRs measured via the point measurement technique

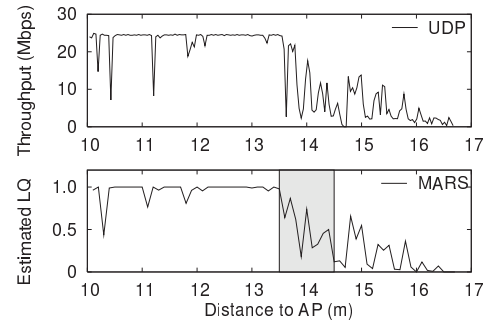


Fig. 4. *Point measurement accuracy*. Link-quality (PDR) measured by MARS (bottom figure) is accurate enough to capture the actual bandwidth (top figure) of a link over a line of 7m from a remote AP.

(lower figure) of a link between MARS and an AP over a straight line of 7m. As shown in the figures, MARS's measurement is indeed close to the actual link bandwidth at the cost of 20 probing packets for a 2s-period at each position. We use the measured PDR (i.e., Estimated LQ in Figure 4 to derive link bandwidth using an PDR-to-link capacity equation in [44]. Here, we use 'link bandwidth' to refer to link capacity. Then, we calculate root-mean-square-error (rmse) between the estimated link bandwidth and achieved UDP throughput. Finally, we derive the accuracy of the measured LQ, based on the average of normalized rmse. Here, this accuracy is derived from each run of the experiment settings as it affects to the accuracy of spatial link-quality estimate. For each run out of the multiple runs, the point measurement technique achieves more than 95% accuracy.

Next, from the point measurements, MARS can also identify a network boundary (gray bar in the lower figure) that may marginally satisfy the link bandwidth. Note that while MARS is capable of measuring PDRs over different data-transmission rates, it uses a fixed-data rate to evaluate the accuracy and applicability of the monitoring technique for spatial measurement protocol, which will be described next.

C. Spatial Measurement Protocol

Next, to characterize link-quality as a function of space, MARS essentially uses a divide-and-conquer approach built upon spatial locality in link-quality. There have been numerous techniques proposed to model spectrum propagation over space. First, empirical models such as log-distance path loss model [38] determine the propagation based on mathematical models and empirical off-line measurements in other similar environments. However, the temporal and spatial granularities of such measurements are too coarse (e.g., retail stores during daytime) to predict the propagation in a heterogeneous deployment environment. Next, ray-tracing-based models can predict the propagation of a signal by tracing rays from a transmitter at uniform angular intervals in all directions [45]. However, this model requires the detailed geometric structure of walls, ceilings, and floors along with information about construction materials. Third, neural-network-based models such as a multi-layer perceptron algorithms have been proposed for cellular networks [46], but they need an extensive training set of terrain information and exhaustive SNR measurements.

By contrast, MARS relies purely on on-line link-quality measurements over unit space without requiring information about physical environments. Specifically, MARS first divides space into grids or *patches* of fixed size (e.g., 50cm × 50cm in an indoor environment). Note that this size changes dynamically, depending on space to probe (an outdoor or corridor environment) and link parameters (transmission power); we use 50cm × 50cm patches and 5 dBm transmission power for our controlled measurements. Next, MARS measures the quality of link between a node and its fixed neighboring node, while changing its location within the patch. These measurements are then averaged to derive spatial link-quality as follows:

$$SPDR_i = \frac{1}{n} \sum_{j=1}^n PDR_j, \quad PDR_j = \frac{1}{m} \sum_{l=1}^m r_l \quad (1)$$

where $SPDR_i$ is the spatial link-quality of patch i , PDR_j is an average PDR of m point measurements at vertex j , and r is the PDR of one point measurement at a given orientation angle. The intuition behind this definition is to exploit spatial locality among the quality of links within a small space [22]. By using a small number of sample measurements within the space, SPDR can represent an average bandwidth of links in the space. Furthermore, as we will describe in Section V, the patch-based definition allows the spatial probing algorithm to selectively and incrementally probe an area of interest, as opposed to probing the entire area. Note that the selection of the patch-size is an interesting issue, which has been explored in [47].

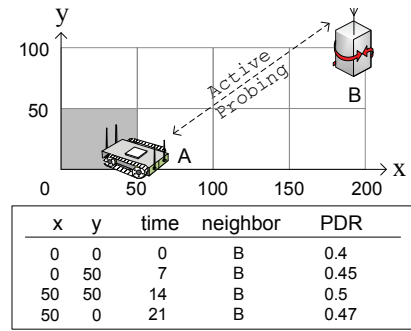
Finally, to reduce temporal and spatial variations in link-quality measurements, MARS not only takes multiple measurements per patch, but also introduces randomization. As shown in Figure 5(a), a MARS node conducts point measurement at each vertex m times. In addition, for each point measurement, the node randomly changes its direction so that measurements can reflect as diverse links as possible. Here, m is dynamically adjusted, depending on the variance of point measurements ($m=3$ by default). MARS also adjusts the number of vertices ($n=4$ by default) within each patch, if the link-quality variance among the vertices is larger than a given threshold. For example, if the variance of PDR is greater than 0.2, MARS linearly increases m and n for the patch. Figure 5(b) shows SPDR measurements in the topology of Figure 5(a). As shown in the figure, SPDR normalizes the quality of links for each patch given a variance constraint (0.1 in PDR), and effectively differentiates spaces with respect to link quality.

V. SPATIAL PROBING ALGORITHM

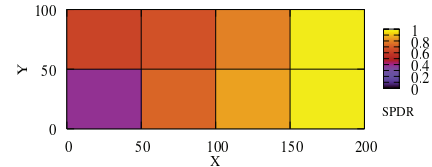
With the measurement protocol, we present a spatial probing algorithm that guides MARS nodes to find locally optimal positions to dynamically (re-)form a string-type relay network.

A. Overview

With a group of nodes, the objective of the spatial probing algorithm in MARS is to *cooperatively* form a relay network by *efficiently* finding a locally *optimal* node position. In other



(a) Spatial LQ measurement protocol



(b) Spatial LQ measurement

Fig. 5. *Spatial link-quality measurement.* (a) MARS measures point link-quality at each vertex of a patch. (b) The point measurement results are then used for deriving SPDR of the patch.

words, the algorithm has to minimize measurement space (equivalent to energy or convergence time) and must identify position that satisfies network QoS requirement. However, main challenges are (i) how to coordinate a group of MARS nodes to efficiently form a relay network and (ii) how to find an ‘interesting’ space for each MARS node to probe. First, coordinating a group of relay nodes is challenging because the link-quality between neighboring nodes depends heavily on each other’s position. Furthermore, this coordination is not an one-time operation, but a recurring operation due to changes in the physical environment or bandwidth demand.

Next, during the formation of a relay network, each MARS node has to efficiently find an *optimal* position at the least measurement cost. Finding an optimal node position in a relay network is essentially equivalent to maximizing the physical distance between adjacent relay nodes, while satisfying the bandwidth demands of their links. On one hand, exhaustive measurements over a target area might be able to provide globally optimal location information, but it might require significant amounts of energy and time. On the other hand, finding a locally optimal position may cause a local maximum, which may result in either poor link-bandwidth or reduced network coverage.

B. Iterative Network (Re-)Formation

For deployment and adjustment of a relay network, the spatial probing algorithm uses an iterative approach for both energy-efficiency and reduction of the coordination overhead. Let us consider the following deployment scenario. Starting from a gateway, Sam, a network administrator, periodically drops a MARS node ($MARS_1, \dots, MARS_n$) along corridors like a trail of breadcrumbs [3], in a way that neighboring MARS nodes can hear heartbeats of each other. After deploying n nodes in a chain, Sam requests the deployed nodes

(e.g., $MARS_i$) to cooperatively optimize their position for meeting the bandwidth (bw) requirement of link to a previous node ($MARS_{i-1}$) or the gateway. One way of coordinating the optimization would be the use of a centralized approach in which each node sends its measurement results to the gateway, and the gateway can calculate the best position of each node. However, this approach requires each node to conduct extensive link-quality measurements over the entire local space. Furthermore, since the bandwidth of one link depends on both end-nodes' locations, the number of measurements that each node conducts increases quadratically. For example, assuming that there are m patches that each node needs to explore, the node has to measure the m patches for each patch of the previous node— $O(m^2)$. Next, a distributed approach would help avoid the need for exhaustive measurements and allow adjacent nodes in the chain to locally identify optimal positions. However, due to the nature of the chain topology in a relay network, even if nodes of one intermediate link locally optimizes their position, the nodes might need to re-adjust their positions after their “parent” or upstream nodes close to the gateway optimize their positions.

By considering such measurement overhead and dependency, the spatial probing algorithm in MARS optimizes its location only after the previous node finds its locally optimal position, based on the iterative approach. As explained in Algorithm 1 (1), the first node ($MARS_1$) optimizes its position with a gateway, and then the child nodes optimize their positions in order. Here, we assume that during the optimization, each node can maintain link connectivity with neighboring nodes, because the heartbeat is transmitted using low-rate, and thus reliable, broadcasts. Because a parent node's location of a link is fixed, a child node needs to measure only m patches over the parent node, which can be further reduced by incorporating a hierarchical approach (see the next section). In addition, the iterative approach facilitates other complex topologies, such as tree or DAG with a linear increase of complexity. For example, once a string relay network is formed, each intermediate node in the network can create another relay network starting from itself, and apply the same iterative procedure to build a tree topology.

In addition to the formation of a relay network, MARS also takes the iterative approach to handle link-quality fluctuations in intermediate links. Each MARS node periodically monitors the quality of link to the next (or child) node, and if that link's quality is below the bandwidth requirement, then the child node should start the adjustment procedure. Subsequently, the child nodes in the remainder of the relay network optimize their position iteratively. We show the effectiveness of this optimization approach in Section VII-B2. Note that this iterative adjustment can be extended for an intermediate node to move and optimize links to both its parent and child nodes in order to avoid the propagation of adjustment requests. This is an interesting, but challenging problem, which is a matter of future inquiry.

C. Hierarchical Position Optimization using Correlation

For each link, a child MARS node optimizes its position by finding a patch whose SPDR meets the bandwidth demand,

Algorithm 1 Spatial probing in MARS

- (1) Main function for formation in $MARS_i$ ($bw, nextloc$)
 - 1: *!** bw : bandwidth demand of link with $MARS_{i-1}$ *!**
 - 2: *!** $nextloc$: location of the next node, $MARS_{i+1}$ *!**
 - 3: wait until node receives *done* message from $MARS_{i-1}$;
 - 4: optimize node location by calling the function (2);
 - 5: send *done* message to $MARS_{i+1}$;
 - (2) Coarse-grained spatial probing ($bw, nextloc$)
 - 6: face toward $nextloc$;
 - 7: **for** $i=0; i < N_g; i++$ **do** *!** N_g is the number of grids *!**
 - 8: move-and-measure corner patches of grid i ;
 - 9: **if** SPDR of the patches is greater than bw **then**
 - 10: $grid \leftarrow i$;
 - 11: **end if**
 - 12: **end for**
 - 13: move to the grid $grid$ and call the function (3)
 - (3) Fine-grained spatial probing ($grid, bw$)
 - 14: $p_o \leftarrow$ current position;
 - 15: **while** SPDR(p_o) satisfies bw **do**
 - 16: move-and-measure neighboring patches of the patch p_o ;
 - 17: $p_m \leftarrow$ the patch with the maximum SPDR among neighbors
 - 18: **if** SPDR(p_m) $> bw$ **then**
 - 19: $p_o \leftarrow$ location of the patch p_m ;
 - 20: **end if**
 - 21: **end while**
-

and further among measured candidate patches, the node chooses the farthest one from its parent node to maximize coverage of networks. In fact, there may exist multiple optimal *positions* that meet the bandwidth demand, and a greedy measurement/movement strategy could lead the node to reach one of the farthest positions. However, relying on a point measurement is often erroneous due to spatially diverse link-quality. Moreover, even after its initial deployment, the robot may need to frequently (re)adjust its position to cope with temporal variations in link-quality. Instead, by using SPDR, MARS positions itself in a patch where the majority of positions meet the demand with a certain variance bound.

A main challenge now is how to “efficiently” find such a patch, as opposed to relying on exhaustive search over the entire deployment area. To overcome this challenge, the spatial probing algorithm in MARS exploits characteristics in spatial link-quality measurements. If the measurements can capture correlations of spatial link-quality with stationary factors such as distance, obstacles, or interference source, then the probing algorithm can adaptively adjust the granularity of measurements.

To confirm this characteristic, we have conducted two interesting measurement studies as follows. First, we study the correlation of SPDRs with distance. We use an empty room (600cm \times 60cm space) in our lab and an idle IEEE 802.11a frequency to exclude the other effects such as interference, obstacles and moving objects. In addition, we use a reduced transmission power of 5dBm to identify the correlation given the limited room space. Note that the results from these controlled settings are also consistent with those in real-life settings, as we will show in Section VII-B1. Next, we place one stationary node and one mobile robot at one corner and measure the spatial link-quality while letting the robot move away from the stationary node. Here, we use an exhaustive

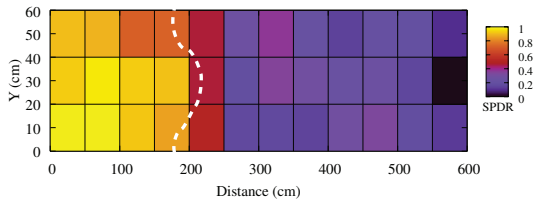


Fig. 6. *Spatial correlation with distance.* Spatial link-quality is correlated with distance between the two end-nodes of a link. MARS can identify the boundary (dotted line) that satisfies bandwidth demand (SPDR = 0.7).

spatial probing algorithm, which measures SPDR of every patch in the area. As shown in Figure 6, the correlation between spatial link-quality and distance is captured. Furthermore, this correlation helps MARS find locations that satisfy the bandwidth demand (dotted line). Second, we study the correlation of SPDRs with physical obstacles as well as the source of signal. We first place an obstacle, a stationary node, and a mobile node as shown in Figure 7(a). While moving toward the end-position, the mobile robot measures and collects SPDR of each patch. Figure 7(b) also confirms that the measurements reflect the strong correlation. These confirmed correlations suggest that the robot may take coarse-grained measurements until the robot reaches *areas* whose SPDRs are close to meeting the bandwidth requirements (correlation with distance). On the other hand, in such interesting *areas*, the robot may need to take fine-grained measurements to optimize its position because of the irregular link-quality distribution resulting from obstacles (correlation with obstacle).

Based on the above observations, our spatial probing algorithm takes a hierarchical approach. As explained in Section IV-C, existing spectrum propagation models either suffer from unmatched physical parameters or require comprehensive information about physical environments to predict optimal node positions. Exhaustive spatial probing requires excessive time and system resources—an area of $5\text{m} \times 5\text{m}$ with a $50\text{cm} \times 50\text{cm}$ unit requires 100 measurements. Instead, MARS takes a two-step hierarchical procedure as explained in Algorithm 1 (2) and (3). In the first step, it divides the probing space using a grid large enough to identify the correlation with distance. Note that the grid size is determined based on the wireless technology used and the environment. We will show one measurement-based grid size for indoors 802.11a in Section VII-B1. Then, MARS measures a subset of patches inside each grid (coarse-grained measurement). Among them, it identifies patches beyond which spatial link-quality does not meet the bandwidth requirements. In the second step, within the grid including the identified patches, MARS uses fine-grained measurements to find a locally optimal location.

Let us consider an example of optimizing one link from an AP. As shown in Figure 8, a MARS node in grid 1 needs to find a location far away from the AP in grid 2. MARS first measures the spatial link-quality of a corner patch ($50\text{cm} \times 50\text{cm}$) in each grid ($5\text{m} \times 5\text{m}$), and then identifies grid 6 as the farthest grid that contains a bandwidth-satisfying patch, shown with a cross in the figure. Next, within the identified grid, MARS uses Newton’s method,

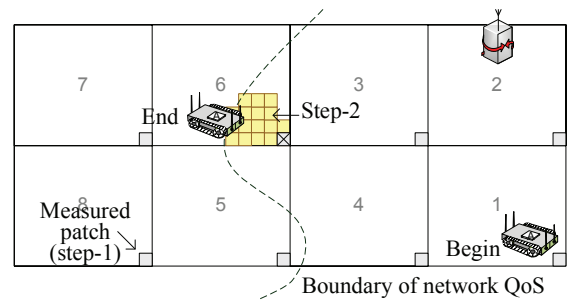


Fig. 8. *Two-step spatial probing procedure of MARS:* MARS first finds the best among 8 grids. Then, within the grid, it identifies a locally optimal patch

which recursively selects the best neighboring patch until the node reaches a locally optimal position. Note that MARS uses Newton’s method for its simplicity, but other optimization methods such as second moments or extrapolation can also be used for fine-grained measurements/movements. Nevertheless, using the hierarchical approach, MARS significantly reduces the probing space, as shown in the example. Our evaluation results in Section VII-B1 also confirm the benefits of the two-step procedure against the exhaustive spatial probing.

VI. POSITIONING SYSTEM

We present the final component of MARS, a positioning system, that provides the location information of a node.

A. Overview

The function of a positioning system is to continuously maintain the accurate location information of a node for both derivation of spatial link-quality and relocation to previous measurement areas. Although it is flexible enough to adopt any positioning system, MARS uses an *infrastructure-less* hybrid positioning system, especially for indoor environments. The system is designed for using dead-reckoning (DR) combined with landmark-based positioning. Although the system deliberately adopts well-known positioning techniques from the robotics, the main purpose of this section is to share our experience in building an inexpensive positioning system tailored for a mobile router, while completing the design and evaluation of MARS.

B. Hybrid Positioning Algorithms

The positioning system in MARS consists of (i) continuous location tracking or dead-reckoning and (ii) periodic landmark-based position measurement. First, DR positioning has been widely used in many navigation systems due mainly to its simplicity. Adopting DR, the positioning system in MARS simply maintains the location information of a robot by constantly updating the robot’s position. Using the robot’s previous position (x) and previous movement information (δ), the system can easily estimate the robot’s current position ($x + \delta$).

However, this technique accumulates errors, due to unexpected obstacles, floor conditions, or physical inertia, which are difficult to avoid. For example, assume that MARS requests

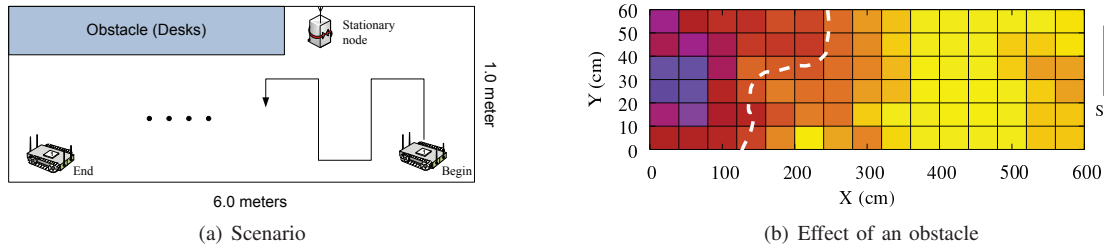


Fig. 7. *Spatial correlation with an obstacle.* Figure 7(b) plots MARS’s measurement result of SPDR in our office (Figure 7(a)). SPDR shows diverse spatial link-quality and correlates with the stationary node and the obstacle. The dotted line also shows MARS’s identification of an interesting network boundary that satisfies the bandwidth requirements.

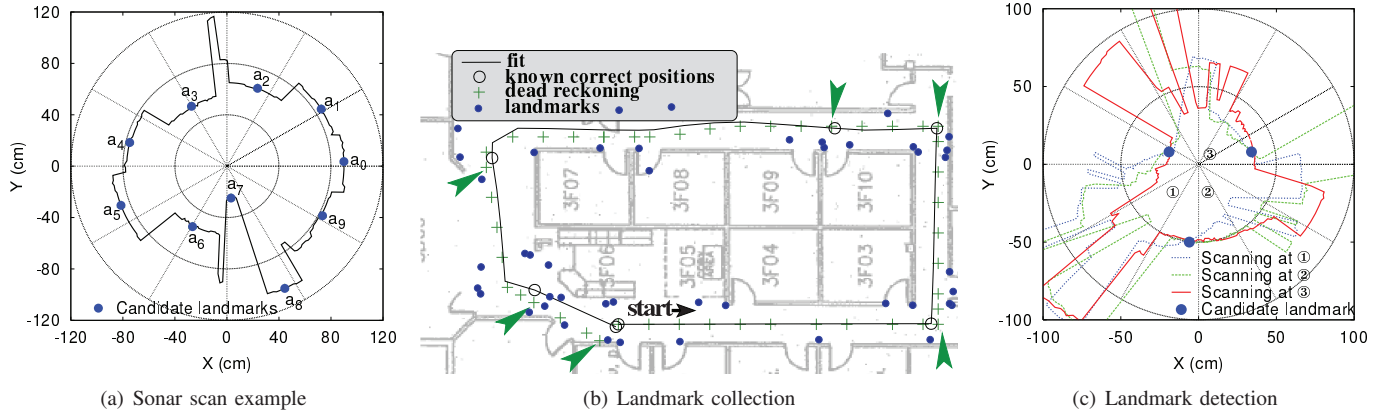


Fig. 9. *Landmark-based positioning:* (a) A sonar scanning primitive identifies landmarks around the robot’s position. (b) MARS discovers landmarks using a semi-automated procedure, which is based on DR and manual measurements of a few positions. (c) MARS determines landmarks that are visible from at least three different locations.

a 90° rotation followed by a movement of 2m but the robot physically rotates 93° instead, while still believing it has rotated 90° . After a forward movement of 2m its physical position is 10cm away from its believed-to-be position. These small errors accumulate over time and may render the robot unable to follow its planned trajectory. While use of a compass would greatly improve the performance of DR which is more sensitive to angular errors, we found that stray magnetic fields (HVAC, power cables, metallic structures) render all compasses unusable indoor.

One method to avoid accumulation of positioning errors is to periodically measure the robot’s true position and update the DR-based position. Numerous position measurement techniques have been proposed, but many require infrastructure support such as sensors and access points [32], [34] or incur extensive computation overhead for processing image/training data [29], [30]. Instead, MARS exploits naturally occurring landmarks in the environment. Briefly, given positions (x_i, y_i) of at least three landmarks and distances d_i to each landmark from the current position, one can derive the current position (x, y) by solving the following equation:

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2. \quad (2)$$

However, the main challenges in using this technique are how to identify the landmarks (x_i, y_i) and how to accurately derive the position in the presence of measurement uncertainties of d_i . First, to sense landmarks near the robot, MARS uses sonar scanning. Spinning around 360° , the robot collects the

information of distances to its surrounding obstacles. Figure 9(a) shows one scan result from the robot located at $(0, 0)$. Because the sonar has a 45° wide beam shape, the resulting polar plot has regions of constant depth (RCDs) only for some of the objects in the environment.

Next, using the above scanning, the positioning system needs to derive the robot’s current position. However, as shown in Figure 9(a), the scan result includes many candidate landmarks (measured as arcs a_0, \dots, a_9), and the robot has to determine which arcs indeed represent landmarks and how these arcs are associated with known landmarks. We assume the position information of landmarks has already been collected, and the next section will discuss how to collect the position information. To solve this association problem, MARS uses a matching algorithm similar to the one in [48]. Briefly, given a set of arcs and known landmarks that are likely to be visible from the robot’s current believed-to-be position, MARS first generates a set of feasible matchings between arcs and landmarks. Then, each feasible matching is evaluated and considered only if from the estimated robot’s position, landmarks are actually sensed at measured distances in the scanning results. Finally, if there are multiple valid matchings, then MARS chooses the matching that minimizes the residual error, defined as follows:

$$\frac{1}{n} \sum_{i=1}^n |(x - x_i)^2 + (y - y_i)^2 - d_i^2| \quad (3)$$

where n is the number of landmarks, (x, y) the estimated position of a robot, (x_i, y_i) the position of a landmark i , and

d_i distance between (x, y) and (x_i, y_i) .

C. Landmark Collection Procedure

To build the landmark information, the positioning system includes a semi-automated landmark collection procedure. This procedure consists of DR-based collection of RCDs over deployment areas, followed by offline processing that extracts landmarks from a large set of collected RCDs. Initially, a robot navigates deployment areas (corridors), and periodically stops and measures RCDs via scanning. These RCDs and the robot’s scanning position information are then used to calculate the positions of potential landmarks. However, for this calculation, accurate information on scanning points is essential, and the robot obtains it in the following semi-automated way. The robot records the scanning positions using DR during the navigation, and a few positions are manually measured. Then, the manual position measurements and the DR-based positions are fed into a trajectory fitting optimization technique for generating true scanning points. Using a few “good” points that are manually measured, the entire trajectory of the robot can be fitted so that the trajectory satisfies two goals: (i) the trajectory passes through good points and (ii) the trajectory is close to the length/angle of each leg from the odometry as much as possible. This trajectory fit is obtained using the following optimization:

$$\min \left[\sum_{i=1}^n d_{err}(i, i+1) + K \sum_{i=2}^{n-1} a_{err}(i-1, i, i+1) \right] \quad (4)$$

where $d_{err}(i, i+1)$ is the absolute difference between the length of leg i obtained from the odometry and the length imposed by the fit; $a_{err}(i-1, i, i+1)$ is the absolute error in angle at measurement stop i , namely, the angle between segments $(i-1, i)$ and $(i, i+1)$; K is a constant that modifies the relative weight of preserving angles versus preserving distances from the original drive. The procedure is implemented using the function *fmins* in *octave-forge*. Finally, based on the true scanning points and collected RCDs, the positioning system can obtain accurate positions of candidate landmarks.

Figure 9(b) depicts the result of the landmark-collection procedure on the corridor of an office building. The robot starts at the circle marked “start”, drives in a counter-clockwise loop, and periodically stops and takes a sonar scan to collect RCDs. At the same time, the robot’s DR-based positions are recorded (denoted by crosses). As expected, DR-based position becomes erroneous toward the end of the drive, where the robot is 60cm away from its DR-based position. On the other hand, using a few manual position measurements (denoted by circles), the robot’s actual trajectory (the solid line) and scanning positions (omitted) are also collected.

Next, based on the information of RCDs and scanning points, the positioning system identifies “good” landmarks that have robust visibility. Good landmarks should be visible from different places, and we found that corners, door frames, or even cracks in the walls are good landmarks mainly because they are fixed and reflective to sonars. MARS only includes landmarks that are visible from at least three scanning points.

Figure 9(c) shows the detection from the scanning results at three different positions denoted as 1, 2, 3. Good landmarks like the top two dots are heavily intersected (high visibility). On the other hand, the candidate landmark at the lowest spot is not “good”, since the intersection is weak—it is in fact a flat wall. Using this technique and RCD collections, MARS effectively collects landmarks as shown in Figure 9(b) that are used later for positioning during the spatial probing. Note that a “good” landmark refers to its presence in measurements from different vantage points, and not its guaranteed presence in time. Clearly, furniture that is later moved would provide unreliable landmarks, which either need to be detected as such, or eliminated by a new collection procedure. The first approach is intensely explored by the robotics community [49], as it provides methods to simultaneously use and update the map. These highly complex methods, based on extended Kalman filters, or on particle filters are able to probabilistically recognize new landmarks to be added to the map, or delete old landmarks which disappeared, all based on the the current known position and the stable landmarks. In this project however, we opted for the latter approach—a lower complexity method that involves the semi-automated landmark collection procedure. This has to be repeated by the robot as needed whenever the set of the landmarks changes.

VII. PERFORMANCE EVALUATION

We now present the evaluation results of MARS. We first describe an experimental setup, and then present key experimental evaluation results in indoor environments. Next, we show *ns-2*-based simulation study on energy saving.

A. Experimental Setup

We extensively evaluated MARS in a challenging indoor environment consisting mainly of office rooms and corridors (Figure 10). This environment includes floor-to-ceiling concrete walls and wooden doors, thus providing natural multi-path fading effects on the radio signal. In addition, IEEE 802.11a is used for wireless links since this standard provides high data-rates and many idle channels. Throughout the entire experimentation a fixed data-rate is used to exclude the effects of rate-adaptation algorithms in NICs and to focus on the effect of node mobility. Each radio is tuned to a medium transmission power of 10 dBm to allow multi-hop relays in a limited space. Finally, we prototype and use three MARS nodes for our experimentation, which are sufficient to demonstrate MARS’ potential benefits for wireless relay networks. By

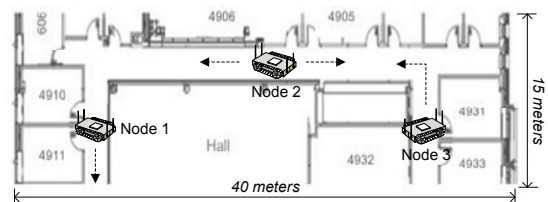


Fig. 10. Topology for corridor experimentation. We evaluate the spatial probing algorithm of MARS in our Department Building.

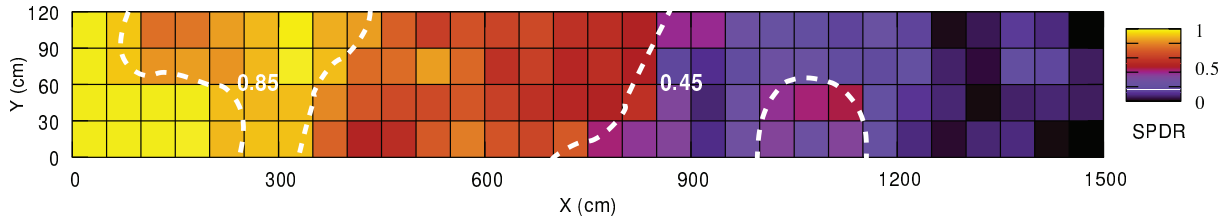


Fig. 11. *Spatial link-quality measurement on a corridor.* Spatial link-quality in a real-life environment also shows correlation with distance and many obstacles. Areas with blue shades and white dot lines are interesting places for MARS to be positioned.

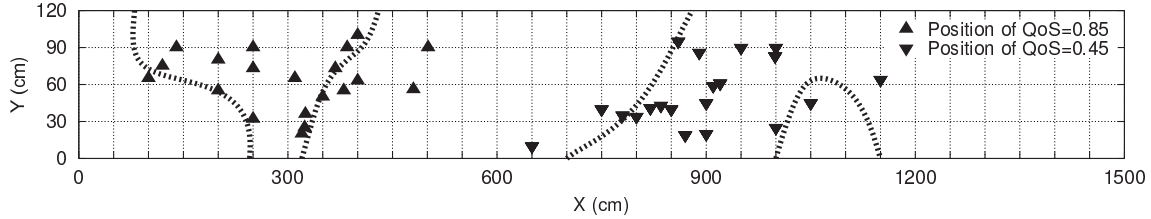


Fig. 12. *Accuracy of spatial probing.* The spatial probing finds locally optimal positions with 86% accuracy for given QoS requests, while reducing the measurement overhead by two-thirds, compared to exhaustive probing.

performing the same experiment recursively, one can realize experimentation on an arbitrary number of hops.

B. Experimental Results

1) *Reducing Space to Probe:* We first studied the effectiveness of the spatial probing algorithm of MARS in reducing space to probe. We place node 1 in one corridor shown in Figure 10 and let node 2 find the position farthest away from node 1 given QoS constraints. We first collect spatial link-quality using exhaustive probing—visit every patch and measure SPDR—over $15\text{m} \times 1.2\text{m}$ on the corridor. Then, we run the spatial probing algorithm explained in Section V-C, 20 times with the same settings. We measure the final position of node 2, the number of point measurements, and the number of patches visited.

Figure 11 shows the spatial link-quality measurements on the corridor using the exhaustive probing. We measure the link-quality on every patch of size $50\text{cm} \times 30\text{cm}$. Note that we ran the same experiment more than 5 times over 2 days and saw similar results. As shown in the figure, the spatial link-quality shows correlation with several obstacles (doors, walls, etc.) along the corridor as well as with distance to the stationary node. In addition, this link-quality shows several interesting boundaries denoted by white lines, or areas (e.g., the one labeled with $\text{SPDR}=0.85$) that satisfy link’s QoS demand. However, even though this exhaustive probing provides a comprehensive SPDR map, it is very expensive in terms of energy and time to build. For example, the above spatial probing for constructing the entire map requires more than 450 point measurements.

The spatial probing algorithm in MARS reduces this overhead while maintaining reasonable accuracy in finding a locally optimal position. Figure 12 shows the distribution of MARS’s final positions from 20 runs for each QoS demand ($\text{SPDR}=0.85$ or 0.45). Starting from $(0, 0)$, the robot is guided by the spatial probing algorithm to find the farthest position that satisfies the given SPDR. As shown in the figure, MARS’s probing algorithm effectively determines a locally optimal

position; 86% of the final locations are located within the area of required SPDR boundary, whereas the remaining 14% deviates from the boundary by, on average, 54 cm. Some of runs yielded local maxima—still satisfying the required QoS, but without reaching the farthest position. MARS could avoid this local maximum by using other optimization techniques, such as extrapolation, in fine-grained measurements, and we will explore these in future. On the other hand, thanks to its hierarchical approach, MARS reduces the number of measurements, on average, by two-thirds over the exhaustive probing, as shown in Table I. For example, $\text{MARS}_{0.45}$ reduces the total number of measurements (N_m) from 465 to 150. Moreover, an interesting feature is that the increase in number of measurements from $\text{MARS}_{0.85}$ to $\text{MARS}_{0.45}$ is only 19%, although the navigation space of the former is twice larger than that of the latter, indicating the scalability of the spatial probing algorithm.

Throughout our experimentation, the spatial probing algorithm is set to use the $2.0\text{m} \times 0.6\text{m}$ size of a grid. This value is determined based on our off-line measurement study, which helps identify the degree of link-quality attenuation in corridor or indoor environments. As increasing the grid size by 1m, we measure the difference between two SPDRs in both ends of the grid, and repeat this measurement until the difference becomes greater than the minimum threshold (10%). Under current experiment settings, the use of a $2.0\text{m} \times 0.6\text{m}$ grid provides 10–100% more efficiency than the use of other’s. This efficiency results from the trade-off in the two-step procedure of spatial probing. For example, the larger the size of grid, the faster the algorithm can find a boundary of interest. However, it is likely that the algorithm needs more fine-grained measurements within a large-size grid.

2) *Optimizing Multi-Hop Links:* We also studied the effectiveness of optimizing multi-hop wireless links in the presence of changes in link conditions. As discussed in Section V-B, to adapt to changing link conditions or QoS requirements, MARS iteratively and cooperatively adjusts its position to maintain the required QoS. To evaluate the performance of this

TABLE I

EFFICIENCY BENEFIT OF THE SPATIAL PROBING IN MARS. MARS REDUCES THE MEASUREMENT EFFORT BY TWO-THIRDS OVER THE EXHAUSTIVE SPATIAL PROBING, WHILE FINDING A LOCALLY OPTIMAL POSITION. THE NUMBER IN A PARENTHESIS IS VARIANCE.

| | Exhaustive | MARS _{0.45} | MARS _{0.85} | Benefits [§] |
|---------|------------|----------------------|----------------------|-----------------------|
| N_m * | 465 (0) | 150 (32) | 126 (43) | 67.3% |
| N_p ¶ | 120 (0) | 24 (6) | 18 (5) | 80.0% |

§ Overhead reduction of MARS_{0.45} over exhaustive probing.

* Total number of measurements.

¶ Total number of patches visited.

adjustment, we place three nodes as shown in Figure 10 along different corridors and let them form multi-hop relay links (node 1 ↔ node 2 ↔ node 3). Next, we move node 1 to south so that the quality of the link between node 1 and 2 is degraded, and we let node 2 and 3 maintain QoS demand (SPDR=0.85). We ran the same experiment 10 times and measured the final positions of node 2 and 3 after their adjustment.

Figure 13 shows the effectiveness of our adjustment in coping with link condition changes. The unshaded part of the figure shows the robots' final positions (denoted as triangles) on the corridor. Starting from their original positions, denoted as N2 and N3, node 2 first starts its spatial probing to adjust its position. Next, once node 2 finds a location that satisfies the QoS requirements (0.85), the node informs node 3 of its adjustment through a broadcast-based message handshake. Then, node 3 starts adjusting its own position with respect to the new location of node 2 in order to maintain the required QoS. As shown in the figure, each robot successfully senses its direction (e.g., east for node 2) without relying on movement information of node 1 and moves in the direction that node 1 moved. In addition, this iterative adjustment significantly reduces the average measurement overhead. Compared to the exhaustive probing where node 2 and 3 have to measure SPDR of every patch in the corridor, the spatial probing in MARS selectively measures SPDR over the fixed previous node. As shown in the inset graph, the iterative adjustment in MARS reduces the total number of measurements ($N_2 + N_3$) by 72% compared to the exhaustive probing-based adjustment (*Ex.*).

3) *Maintaining Positioning Accuracy:* We now turn to the quantitative evaluation of MARS's positioning system. We first set up one large square space (240 cm × 240cm) with four artificial landmarks in our lab. In the square, we let the robot collect positions of the four landmarks through the landmark collection technique described in Section VI-C. As expected, the robot properly detects all the landmarks with error of at most 2cm from their true positions, and this measured landmark information is used for the robot's positioning during the following random walks. At each point on the random walk, the robot updates its position (i.e., believed-to-be position) based on its previous position and movement information. At the same time, the robot takes a sonar scan (as in Figure 9(a)) and derives its current true position using the scanned result. Next, the robot derives the position error by comparing the believed-to-be position and the true position from the scan. Finally, the robot updates its position information with the

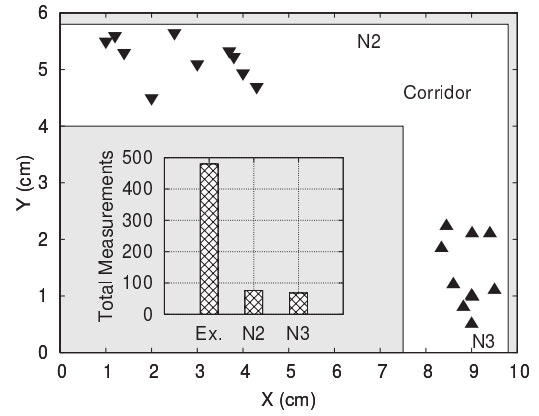


Fig. 13. The multi-hop optimization result. In MARS, relay nodes optimize each position one-by-one when the link conditions changed. Triangles denote adjusted nodes' positions.

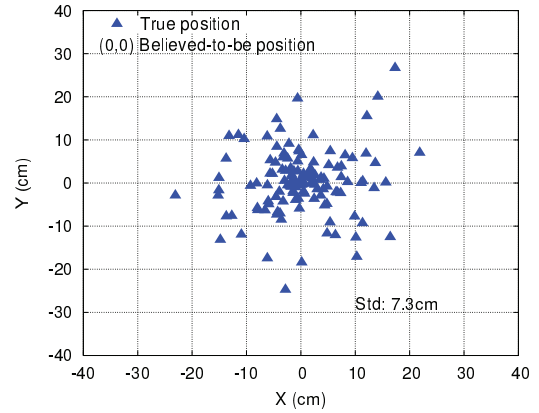


Fig. 14. Accuracy in correcting location error. The positioning system in MARS accurately corrects the error in its location information, on average, within 7.3cm error.

true position and drives itself toward the next random point. Figures 14 shows the robot's true positions relative to believed-to-be positions (0, 0). As shown in the figure, the MARS's positioning system keeps the average location error less than 7.3cm, thanks to landmark-based position measurements. In addition, the distributions of errors is found to be independent Gaussians on x and y with an almost diagonal covariance matrix. This error is due mostly to the drifts caused by the heading error, but it is small enough for the robot to correct the error using the landmarks and continue its walk without unbounded growth of error.

C. Simulation

We also evaluated the effectiveness of MARS in energy savings by using the *ns-2* simulator [12]. We implemented MARS as a mobile node in *ns-2*. We used the wireless extension for 802.11 MAC in *ns-2* and used a string-type relay topology with three nodes as in our previous experiments for comparison. While reducing the unit grid size of the spatial probing algorithm, we measured the total energy consumption during the relay network (re-)formation. Note that the smaller the grid size, the higher reformation accuracy MARS can achieve. The reformation is triggered by changing the link-

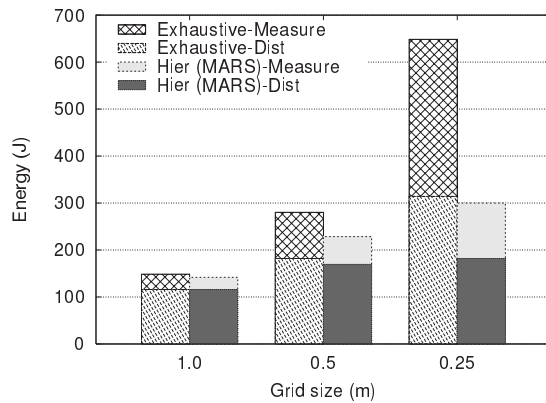


Fig. 15. MARS’s energy savings. The spatial probing algorithm in MARS saves energy by reducing the navigation distance as well as the number of measurements, compared to the exhaustive measurements.

quality requirement (from 0.75 to 0.85) of the link between the first and second nodes. Then, the second and third nodes sequentially relocate themselves to meet the requirements. To calculate energy consumption, we use unit energy levels for (i) robot’s moving distance (i.e., 8.27 J per meter) and (ii) link-quality measurement (2.184 J per vertex), both of which are derived from measurements in [50], [51] together with the MARS measurement protocol. Here, we do not consider the energy consumption for operating a wireless router board, since the wireless router is assumed to be always on for network connectivity. For comparison, we also implemented and used an exhaustive measurement strategy.

Figure 15 shows MARS’s energy savings per node compared to the exhaustive strategy. As shown in the figure, the MARS probing algorithm saves energy by up to 54% (for the grid size of 0.25 m). In addition, as the grid size decreases, the energy savings comes from the reduced navigation distance (i.e., Hier-Dist) as well as the reduced number of measurements (Hier-Measure) achieved by the MARS’s hierarchical probing algorithm.

VIII. CONCLUSION

A. Concluding Remarks

In this paper, we have presented MARS—a mobile wireless router that is aware of spatial diversity in wireless link-quality. MARS autonomously measures spatial wireless link-condition and reforms a wireless relay network with neighboring nodes. We have built MARS’ prototype using commodity robots and IEEE-based wireless routers. Using extensive experimental evaluation, we have demonstrated the feasibility and practicality of MARS for dynamic (re)formation of a multi-hop relay network. MARS demonstration videos are available in a public website of <http://kabru.eecs.umich.edu/bin/view/Main/SMART>.

B. Remaining Issues

While the current prototype targets 802.11-based indoor applications, MARS can be extended further by addressing the following issues:

- *Flexible deployment scenarios:* Even in cases when routers are deployed by humans [3], MARS can support the deployment scenario via iterative adjustments around each drop point. This is especially useful for inherently hazardous scenarios such as rescue or military applications, but a robotic platform that is more capable than the iRobot is necessary for such applications.
- *Fully automatic landmark collection:* The current landmark collection in MARS requires a few manual position measurements. This may be time-consuming and not scalable in large networks. We plan to investigate ways to fully automate this procedure [31].
- *Transmission rate adaptation:* In this paper, MARS focuses on the mobility of a wireless router and disables the use of link-layer rate adaptation (i.e., fixed rate). However, it would be an interesting to jointly consider transmission rate adaptation (e.g., [6]), which we plan to investigate as a separate paper.

ACKNOWLEDGEMENT

This work was partially supported by European Social Fund in Romania, grant POSDRU/88/1.5/S/62557.

REFERENCES

- [1] PacketHop Inc., “Mobile Router,” <http://www.packethop.com>.
- [2] Rajant Corp., “BreadCrumb XL,” <http://www.rajant.com>.
- [3] M. R. Souryal, J. Geissbueheler, L. E. Miller, and N. Moayeri, “Real-time deployment of multihop relays for range extension,” in *Proceedings of ACM MobiSys*, San Juan, Puerto Rico, Jun. 2007.
- [4] J. Camp, J. Robinson, C. Steger, and E. Knightly, “Measurement driven deployment of a two-tier urban mesh access network,” in *Proceedings of ACM MobiSys*, Uppsala, Sweden, Jun. 2006.
- [5] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, “Opportunistic media access for multirate ad hoc networks,” in *Proceedings of ACM MobiCom*, Atlanta, GA, Sep. 2002.
- [6] S. Wong, H. Yang, S. Lu, and V. Bharghavan, “Robust rate adaptation for 802.11 wireless networks,” in *Proceedings of ACM MobiCom*, Los Angeles, CA, Sep. 2006.
- [7] A. M. Sayeed and V. Raghavan, “Maximizing MIMO capacity in sparse multipath with reconfigurable antenna arrays,” *IEEE Journal on Special Topics in Signal Processing*, 2007.
- [8] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, “A multi-radio unification protocol for IEEE 802.11 wireless networks,” in *Proceedings of IEEE BroadNets*, San Jose, CA, Oct. 2004.
- [9] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, “Link-level measurements from an 802.11b mesh network,” in *Proceedings of ACM SIGCOMM*, Portland, OR, Aug. 2004.
- [10] D. S. D. Couto, D. Aguayo, B. A. Chambers, and R. Morris, “Performance of multi-hop wireless networks: Shortest path is not enough,” in *Proceedings of the First Workshop on Hot Topics in Networks (HotNets-1)*, Princeton, New Jersey, Oct. 2002.
- [11] W. Xu, T. Wood, W. Trappe, and Y. Zhang, “Channel surfing and spatial retreats: Defenses against wireless denial of service,” in *Proceedings of ACM WiSe*, Philadelphia, PA, Oct. 2004.
- [12] The Network Simulator ns-2, <http://www.isi.edu/nsnam/ns>.
- [13] J. Robinson and E. Knightly, “A performance study of deployment factors in wireless mesh networks,” in *Proceedings of IEEE InfoCom*, Anchorage, AL, May 2007.
- [14] I. Akyildiz, X. Wang, and W. Wang, “Wireless mesh networks: A survey,” *Computer Networks*, no. 47, pp. 445–487, 2005.
- [15] R. Chandra, L. Qiu, K. Jain, and M. Mahdian, “Optimizing the placement of integration points in multi-hop wireless networks,” in *Proceedings of IEEE ICNP*, Berlin, Germany, Oct. 2004.
- [16] DARPA/IPTO, “LANdroids: Autonomous, expendable, threat detection and locating system,” <http://www.darpa.mil/ipto>.
- [17] A. Raniwala and T.-C. Chiueh, “Deployment issues in enterprise wireless lans,” Technical Report, Experimental Computer Systems Lab, State University of New York, Tech. Rep. TR-145, Sep. 2003.

- [18] Y.-C. Cheng, J. Bellardo, P. Benko, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: Solving the puzzle of enterprise 802.11 analysis," in *Proceedings of ACM SIGCOMM*, Pisa, Italy, Sep. 2006.
- [19] K.-H. Kim and K. G. Shin, "On accurate measurement of link quality in multi-hop wireless mesh networks," in *Proceedings of ACM MobiCom*, Los Angeles, CA, Sep. 2006.
- [20] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Analyzing the mac-level behavior of wireless networks in the wild," in *Proceedings of ACM SIGCOMM*, Pisa, Italy, Sep. 2006.
- [21] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin, "Temporal properties of low power wireless links: Modeling and implications on multi-hop routing," in *Proceedings of ACM MobiHoc*, Illinois, 2005.
- [22] Y. Xu and W.-C. Lee, "Exploring spatial correlation for link quality estimation in wireless sensor networks," in *Proceedings of IEEE PerCom*, Pisa, Italy, Mar. 2006.
- [23] T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu, "Preemptive routing in ad hoc networks," in *Proceedings of ACM MobiCom*, 2001, pp. 43–52.
- [24] S. Chellappan, X. Bai, B. Ma, and D. Xuan, "Sensor networks deployment using flip-based sensors," in *Proceedings of IEEE MASS*, Washington, DC, Nov. 2005.
- [25] W. Wang, V. Srinivasan, and K. C. Chua, "Trade-offs between mobility and density for coverage in wireless sensor networks," in *Proceedings of ACM MobiCom*, Montreal, CA, Sep. 2007.
- [26] M. Lindhe and K. H. Johansson, "Using robot mobility to exploit multipath fading," *IEEE Wireless Communications*, Feb. 2009.
- [27] Q. Li and D. Rus, "Sending messages to mobile users in disconnected ad-hoc wireless networks," in *Proceedings of ACM MobiCom*, Boston, MA, Aug. 2000.
- [28] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of adhoc wireless networks," *IEEE Transactions on Information Theory*, vol. 10, no. 4, Aug. 2002.
- [29] J. Campbell, R. Sukthankar, I. Nourbakhsh, and A. Pahwa, "A robust visual odometry and precipice detection system using consumer-grade monocular vision," in *Proceedings of IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 2005.
- [30] D. M. Pierce, "Map learning with uninterpreted sensors and effectors," *Doctoral dissertation, Department of Computer Sciences, The University of Texas at Austin*, 1995.
- [31] S. Thrun, "Robotics mapping: A survey," Technical Report CMU-CS-02-111, School of Computer Science, Carnegie Mellon University, Tech. Rep., Feb. 2002.
- [32] P. Bahl and V. N. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *INFOCOM (2)*, 2000, pp. 775–784.
- [33] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *ACM MOBICOM*, Boston, MA, 2000.
- [34] P. De, R. Krishnan, A. Raniwala, K. Tatavarthi, N. A. Syed, J. Modi, and T. Chiueh, "Mint-m: An autonomous mobile wireless experimentation platform," in *Proceedings of ACM MobiSys*, Uppsala, Sweden, 2006.
- [35] iRobot Corp., <http://www.irobot.com>.
- [36] J. Reich, V. Misra, and D. Rubenstein, "Spreadable connected autonomic networks," CUCS-016-08, Columbia University, Tech. Rep., 2008.
- [37] Router Board, <http://www.routerboard.com>.
- [38] T. S. Rappaport, "Wireless communications: Principles and practice," Second Edition, Pearson Education, Dec. 2001.
- [39] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-based models of delivery and interference in static wireless networks," in *Proceedings of ACM SIGCOMM*, Pisa, Italy, 2006.
- [40] D. S. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of ACM MobiCom*, San Diego, CA, Sep. 2003.
- [41] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of ACM MobiCom*, Philadelphia, PA, Sep. 2004.
- [42] D. Giustiniano and G. Bianchi, "Are 802.11 link quality broadcast measurements always reliable?" in *Proceedings of ACM CoNEXT*, Lisboa, Portugal, Dec. 2006.
- [43] K.-H. Kim and K. G. Shin, "Accurate and asymmetry-aware measurement of link quality in wireless mesh networks," *To appear in IEEE/ACM Transactions on Networking*, 2009.
- [44] K. Kim and K. Shin, "Self-reconfigurable wireless mesh networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, Apr. 2011.
- [45] K. R. Schaubach, N. J. Davis, and T. Rappaport, "A ray tracing method for predicting path loss and delay spread in microcellular environments," in *Proceedings of IEEE Vehicular Technology Conference*, May 1992.
- [46] A. Nesovic, N. Neskovic, and D. Paunovic, "Macrocell electric field strength prediction model based upon artificial neural networks," *IEEE JSAC*, vol. 40, no. 6, Aug. 2002.
- [47] K.-H. Kim, A. Min, and K. Shin, "Sybot: An adaptive and mobile spectrum survey system for wifi networks," in *Proceedings of ACM MobiCom*, Chicago, IL, Sep. 2010.
- [48] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, pp. 376–382, Jun. 1991.
- [49] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii state of the art," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 3, pp. 108–117, 2006.
- [50] M. Rahimi, H. Shah, G. S. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," in *Proceedings of IEEE International Conference on Robotics and Automation*, Sep. 2003.
- [51] I. Atheros Communications, "Power consumption and energy efficiency comparisons of wlan products," *White paper*, 2003.



Kyu-Han Kim received the BS from the Korea University, Seoul, Korea, the MS from the from the Georgia Institute of Technology, Atlanta, GA, and the PhD from the University of Michigan, Ann Arbor, MI, all in computer science. He is currently a senior research scientist at Hewlett-Packard Laboratories, CA. His research interests include performance, Quality-of-Service, and manageability in mobile/distributed systems and wireless networks. He is a recipient of the ACM MobiCom Best Student Paper Award (Co-author, 2003) and the government scholarship (2001–2005) from the Ministry of Information and Communication, Republic of Korea.



Kang G. Shin is the Kevin & Nancy O'Connor Professor of Computer Science in the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor. His current research focuses on computing systems and networks as well as on embedded real-time and cyber-physical systems, all with emphasis on timeliness, security, and dependability.

He has supervised the completion of 71 PhDs, and authored/coauthored about 800 technical articles (about 300 of these are in archival journals), one a textbook and more than 20 patents or invention disclosures, and received numerous best paper awards, including the Best Paper Awards from the 2011 ACM International Conference on Mobile Computing and Networking (MobiCom11), the 2011 IEEE International Conference on Autonomic Computing, the 2010 and 2000 USENIX Annual Technical Conferences, as well as the 2003 IEEE Communications Society William R. Bennett Prize Paper Award and the 1987 Outstanding IEEE Transactions of Automatic Control Paper Award. He has also received several institutional awards, including the Research Excellence Award in 1989, Outstanding Achievement Award in 1999, Distinguished Faculty Achievement Award in 2001, and Stephen Attwood Award in 2004 from The University of Michigan (the highest honor bestowed to Michigan Engineering faculty); a Distinguished Alumni Award of the College of Engineering, Seoul National University in 2002; 2003 IEEE RTC Technical Achievement Award; and 2006 Ho-Am Prize in Engineering (the highest honor bestowed to Korean-origin engineers).



Dragoș Niculescu received a B.E. from Politehnica University of Bucharest in 1996, and a Ph.D. from Rutgers University in 2004, both in computer science. He then spent five years as a researcher at NEC Laboratories America in the Mobile Communications and Networking Research group working on wireless networking related problems. He is currently an assistant professor at ETTI, University Politehnica of Bucharest, Romania, teaching and researching mobile networking, wireless meshes, and software defined radios.