MagicLink: Weaving Multi-site Wireless Sensor Networks for Large-Scale Experiments

Xinxin Liu[1], Li Yu[2], Di Wang[1], and Xiaolin Li[1]

[1] Scalable Software Systems Laboratory
University of Florida, Gainesville, FL 32611, USA
xinxin@cise.ufl.edu, wdiyou@ufl.edu, andyli@ece.ufl.edu
[2] Google, Moutain View, CA 94043, USA
yyuy@cs.okstate.edu

Abstract. Despite the promising vision of pervasive sensor networks of thousands of nodes, conducting such large-scale experiments on demand is still far from reality due to the limitations of resources, space, and maintenance. To address such challenges, we propose the MagicLink middleware to "magically" weave geographically distributed sensor networks into a large-scale sensor network testbed. MagicLink is a key part of the OKGems remotely programmable cyber-physical system project under the GENI (Global Environment for Network Innovation) initiative; and MagicLink is designed to enable shared "clouds" of sensors for sensor network research and experiments at scale and on demand. Specifically, MagicLink has the following salient features: (1) seamless integration of multi-site sensor networks offering elastic and scalable testbeds; (2) online adaptive simulation that adopts a realistic radio model making the cross-site Internet connection behave like a one-hop sensor network link in real environment; (3) component-based design allowing easy integration with user applications. To the best of our knowledge, MagicLink is the first solution to enable "almost-real" large-scale sensor network experiments across sites. In this paper, we present MagicLink's system architecture and subsystem design. We demonstrate the usability and fidelity of MagicLink through experimental results with representative applications on a two-site testbed.

1 Introduction

The rapid evolution of ubiquitous sensing and pervasive computing paradigm has spurred increasing demand for Wireless Sensor Networks (WSNs) that consist of thousands of sensor nodes and span over large geographic territories. Due to the high cost of sensor motes and lack of testing environment, simulation has been the primary method for researchers to test their sensor network protocol and application design before actually deploying it on a real system. Although simulators provide users the flexibility of acquiring various sizes of virtual networks on demand and reproducing their experiments under different settings, they cannot provide satisfactory performance for both large-scale and high fidelity sensor network simulations. On one hand, some unrealistic abstractions used in simulation hinders the studying of protocol performance under real world constraints. On the other hand, high-fidelity simulation of every event on a sensor mote dissipates computation resources, thus makes it extremely unscalable [1].

T. Korakis, M. Zink, and M. Ott (Eds.): TridentCom 2012, LNICST 44, pp. 194–209, 2012.

With the accelerating development of WSNs, using experimental testbeds for prototyping and validating protocols and applications becomes a preferred method, because it allows for investigating program performances in diverse hardware platforms and various environmental settings. There has been an explosive deployment of sensor network testbeds in the past few years. Numerous small scale testbeds, typically from 20 to 40 sensor nodes, have been deployed and used in many research laboratories. However, the scales of these individual testbeds are insufficient for future WSN applications. To work around this issue, one alternative is to remotely deploy experiments on publicly accessible large-scale sensor platforms, e.g., MoteLab [2], MistLab [3], KanseiGenie [4] and NetEye [5]. Users access sensor resources through assigned accounts and program the sensor motes using customized end-to-end programming tools. However, these platforms suffer from two major problems: First, the scale of such platform is limited to a single site, and is hard to be extensible in terms of resource federation. Second, the provider-dependent interfaces and data logging methods at various platforms hinder the users from reusing their programs. Thereafter, there is a urgent demand for large scale and high fidelity experimental testbeds.

Some recent studies have proposed to create federated sensor networks across multiple sites [4, 6–8]. However, these federated testbeds lack high-fidelity radio models for virtual links across sites. Although these platforms offer connections for data collections and disseminations, they are not suitable for large-scale experiments across multiple sites for protocol design that require realistic radio properties in the virtual links across sites.

We propose the MagicLink system as a hybrid approach to overcome both deficiencies of inaccurate radio abstraction in simulators and inflexible sensor resource acquisition in testbeds. MagicLink aims at fully utilizing isolated sensor resources to enable large-scale sensor network experiments. To accomplish this goal, MagicLink first establishes message tunnels on top of the intermediate Internet connections of the distributed sensor resources. Then, MagicLink restores the original radio link properties over the Internet connections by employing an adaptive radio model to highlight the essential the features of wireless radio links in sensor networks.

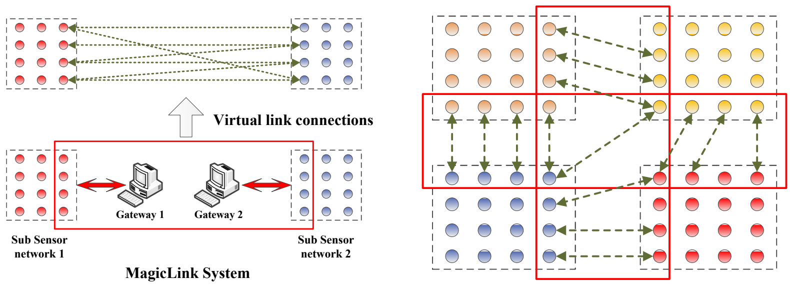
The following design features make MagicLink stand out from other projects:

1. MagicLink is highly elastic and scalable in that it weaves isolated small sensor networks. The federated testbed provides not only data sharing, but also means for cross-layer network protocol design and testing.
2. MagicLink features a unique adaptive radio model that preserves the lossy, anisotropic, and dynamic properties of a real radio link in sensor networks deployed in real world environments.
3. MagicLink is component-based, thus is easy to customize and integrate with user applications.

The rest of this paper is organized as follows. Section 2 defines scope and presents system architecture overview. Section 3 elaborates on the core component of MagicLink. Section 4 presents resource initialization and updating functionalities. Section 5 illustrates cross-site communication details. Section 7 summarizes state-of-the-art related works. Finally, Section 8 concludes this paper and discusses potential future work.

2 MagicLink Design

The motivation behind MagicLink is to unite massive small-scale sensor testbeds prevalent in many research laboratories for large-scale experiments. Sensor nodes in these small scale platforms usually have continuous power supply for long-term usage and connections to computers for easy reprogramming. The rationale of MagicLink is analogous to cloud computing, or peer-to-peer desktop systems [9, 10], which harvest geographically distributed computing resources for computationally intensive applications. Similarly, MagicLink enables a "cloud" of sensor networks that federates multi-site of sensor resources and presents itself as a single large sensor network testbed. Through testbed federation, MagicLink enables researchers to access to diverse resources that are hard to acquire in a single site deployment and can also include and mix real world sensor networks deployed for precision agriculture [11, 12] and habitat monitoring [13]. It can be viewed as a middleware to jointly connect physically separated small sensor networks together, as shown in Figure 1(a). Challenges, however, arise not only in the process of building up connections among the gateway computers of these sensor testbeds, but also in smoothing the Internet gaps to achieve near identical radio link features in a real large-scale sensor testbed.



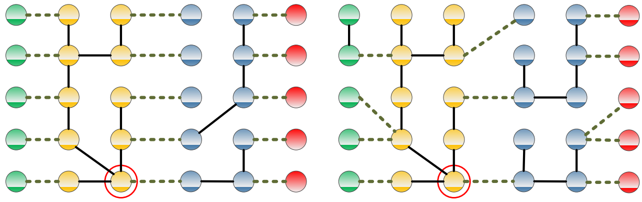
(a) Illustration of construction of virtual links over wired connection.

(b) Large-scale testbed constructed by MagicLink System.

Fig. 1. MagicLink system illustration. Colored circles represent sensor nodes; squared areas indicate geographically separated testbeds. Using MagicLink for sensor resource federation results in a large-scale, smoothly connected sensor testbed that includes real and virtual radio links.

MagicLink primarily addresses two challenges: (1) **Scalable resource federation**: the system should be able to support federation of many sub sensor networks to form a large scale experiment platform. (2) **High-fidelity one-hop virtual link connection simulation**: to construct a seamless connection, the simulated virtual links should hide Internet connection features, e.g., packet loss ratio and round trip delay, but simulate high-fidelity characteristics of radio links in a sensor network. Further, the interfacing functionality should mimic radio communication properties for investigating protocol behaviors in real world settings.

Use Case: To better illustrate the necessity of the high-fidelity virtual link connection simulation, let us consider the scenario of running the Surge application [1] that builds a



(a) Routing tree for Surge without adaptive radio model
(b) Routing tree for Surge with adaptive radio model

Fig. 2. Routing tree for the Surge application on a federated testbed of four sites. Each sub sensor network is highlighted with different colors. Double circled nodes represent root nodes, black solid lines represent radio connections, and green dashed lines represent virtual links across sites. (a) Without adaptive radio model, Internet-based virtual links are always preferred over real radio links as next-hop in the routing path. (b) With adaptive radio model, Internet-based "magic" virtual links and real radio links are treated similarly. Either a magic link or a radio link is selected as next-hop based on their runtime link qualities. This results in a real large testbed rather than simply "connected" but actually "partitioned" multiple testbeds.

routing tree on a federated testbed consisting of four sites. A sensor node running Surge will actively probe its neighbors and select the one with the most reliable link as its routing path. Figure 2(a) depicts the resulting tree on a rigidly connected four-site testbed. In this case, no radio model is applied and the raw Internet connections are used. Since Internet connections are more reliable than radio links (especially when certain reliable transmission protocol such as TCP is adopted), they are most likely to be chosen as the routing path. This is undesirable because it lacks the essential characteristics, e.g., heterogeneity and dynamics, of real radio links. Without these features, the reliability, robustness, and self-adaptive qualities of a protocol cannot be thoroughly investigated. On the other hand, MagicLink employs an adaptive radio model to connect testbeds. As shown in Figure 2(b), a more realistic routing tree is constructed, because the intermediate Internet connections preserve the essential features of radio links. Hence, protocol reliability and robustness can be better investigated.

To ease our presentation, we use the following terminologies throughout the rest of this paper:

– *Sub Sensor Network (SSN)*: a small sensor testbed that are federated into MagicLink.
– *Edge nodes*: those sensor nodes that are at the edge of a SSN.
– *Virtual links*: the virtual connections among the edge nodes from different SSNs.
– *SSN Gateway*: a computer that provides Internet connection interface of a SSN.

System overview: The design of MagicLink system is centered around a set of virtual link specifications called *virtual link pool*, as shown in Figure 3. All the cross-site connection information is maintained in this virtual link pool. Four types of operations are performed on the virtual link pool to achieve distributed resource weaving: *system initialization*, *system monitoring*, *virtual link adaptation*, and *message dispatching* as illustrated in Figure 3.

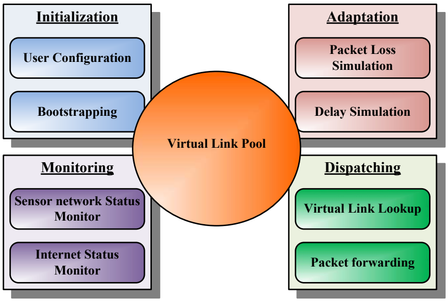


Fig. 3. Component based design of MagicLink System

Virtual link pool contains the information about each individual link constructed by MagicLink system. In its simplest form, it is a list of all source and destination sensor node pairs and their properties, such as transmission delay and packet reception rate. The following key functionalities are implemented:

- Interact with users for testbed configuration and result retrieving
- Locate distributed sensor resources (e.g. SSNs) and establish a connections between their gateway computers
- Build virtual links for each edge sensor node with appropriate setups
- Provide status monitoring service for updating virtual link properties
- Dispatch cross-site messages to their destination
- Customize communication methods for edge nodes to transmit to both radio and wired interfaces

These functionalities are encapsulated in different components and are decoupled from each other as shown in Figure 3. Such a component-based design offers flexibility and easy integration with user applications. The basic functionalities are wrapped into four components: *resource initialization*, which includes (1) interacting with users for system configuration; (2) bootstrapping the system and initializing various parameters for virtual link pool; *monitoring* of resources, which involves (1) keeping tracking locally connected sensor nodes and (2) establishing and maintaining remote gateway computer connections; *message dispatching*, which responsible for (1) providing joint messaging interface between radio links and Internet, (2) performing virtual link lookup and forwarding packets to appropriate virtual links; *adaptation*, which indicates the operations for emulating radio link behavior by adaptively adjust message dispatching operations on virtual links.

3 Virtual Link Pool

Virtual link pool is the central component of MagicLink and maintains virtual links' information. Its primary purpose is to manage edge nodes' communication radius as well as connection qualities. Each virtual link between a pair of edge nodes is represented as a source/destination pair with certain link quality metrics in the virtual link pool. Since radio links are inherently asymmetric, individual gateway computer maintains a separate virtual link pool connecting its local edge nodes to remote edge nodes. The quality of a virtual link is measured in terms of packet reception rate and transmission delay. As MagicLink's target is to seamlessly weave multi-site sensor networks into one large-scale experiment platform, an adaptive radio link model is essential for determining virtual link properties in cross-site message transmission.

It is well-known that radio communications are irregular and unpredictable. To construct a realistic radio model in MagicLink, we not only consider the property of **radio signal path loss and shadow fading**, but also further incorporate a link quality adjustment component to reflect the **anisotropic transmission** and **dynamically changing** property of a radio link. For initial link quality setup, MagicLink uses analytical model to calculate packet reception rate at user specified distance and transmission power. During the execution of the user application, virtual link qualities are periodically updated in accordance to the edge node's real link qualities.

Radio signal path loss and shadow fading: There has been extensive studies on radio propagation under different constraints [14]. In MagicLink, two radio signal path loss models, namely free-space model and two-ray ground model, are provided for ideal line-of-sight communication and single ground reflection scenarios. Users can determine the virtual link distances and choose the desired environment parameter settings for their experiment.

Besides radio signal attenuation, noise on wireless channels as well as other shadowing factors, such as reflecting and scattering, also result in degradation in received signal strength. The most commonly used statistic model for describing this shadowing effect is *log-normal* distribution [14–17]. Therefore, the final received power P_f is the summation of the attenuated transmission power P_r and the shadowing effect, and it also follows the *log-normal* distribution $P_f \sim Logn(P_r, \sigma)$. We use the subscript "dB" to indicate the decibel form of a variable from now on, thus the final received power $P_{f\text{dB}} \sim \mathcal{N}(P_{r\text{dB}}, \sigma_{\text{dB}})$. The variance σ_{dB} of this distribution is a environment related factor, and it is preset by the users to represent the characteristics of different environments.

MagicLink uses a threshold value ξ to infer the correlation between packet reception and final received transmission power. Based on the threshold, we can calculate the probability of a packet being successfully received using:

$$p(P_{f\text{dB}} > \xi_{\text{dB}}) = Q(\frac{\xi_{\text{dB}} - P_{r\text{dB}}}{\sigma_{\text{dB}}}), \tag{1}$$

where the Q-function is defined as the probability that a Gaussian random variable X with mean 0 and variance 1 is greater than certain value.

Anisotropic radio propagation: Anisotropic radio propagation is another significant property to model radio transmission. It should be carefully preserved when building up virtual link connections between two SSNs. Many modeling methods have been proposed in previous studies, e.g., [18], herein we describe the anisotropic property of a sensor node's radio transmission in terms of the *degree of irregularity* metric defined in [18]. The degree of irregularity parameter ϑ of a node is defined as *the maximum path loss percentage variation per unit degree change in the direction of radio propagation*, and is used to calculate the virtual link status based on its relative digression from a predefined direction. The value of ϑ is typically a small number (e.g., 0.005) and it is preset by users for desired radio irregularity degree. Using the metric of irregularity degree requires that the testbed topology is preconfigured, and all the gateway computers know the network topology in advance. This information can be easily obtained via bootstrapping phase. Without the knowledge of testbed topology, a pre-calculated ϑ value can also be obtained from a radio's specification sheet (e.g., Telosb's specification sheet [19]).

Once ϑ is set, the theoretical received transmission power between two edge sensor motes at a relative angle can be adjusted using:

$$P'_r = (1 \pm \vartheta) \times P_r. \tag{2}$$

With Equation 1 and 2, and using dB form of the final adjusted received transmission power P'_f as previously mentioned, the theoretical packet delivery ratio P_{prr} of a virtual link can be calculated as:

$$P_{prr} = p(P'_{f\mathrm{dB}} > \xi_{\mathrm{dB}}) = Q(\frac{\xi_{\mathrm{dB}} - P'_{r\mathrm{dB}}}{\sigma_{\mathrm{dB}}}). \tag{3}$$

Using the radio propagation models enables the users to test their algorithms under different settings, and provides a starting point for virtual link simulation. However, the general free-space and two-ray models may not be able to accurately describe the radio link path loss or fading parameters due to reasons such as occasional obstacles or temporary interferences from other sources. Hence, in MagicLink design, we further adjust the packet reception ratio P_{prr} using the measured data.

Dynamically changing radio link quality: Since wireless links are extremely sensitive to environmental changes, capturing the dynamically changing behavior of a wireless link becomes an indispensable task for link simulation. Although the causes of a link quality fluctuation can be complex, we observed that when all radio links surrounding one sensor node exhibit sudden changes of packet reception ratio, it is highly possible that problem occurs at that sensor node, i.e., low battery level or presence of physical obstacles in the near vicinity. Based on this observation, we can safely infer that the virtual link should also be affected and adjust its link quality accordingly. In order to make our model reflect this temporal property of a link, we introduce a link quality coefficient ψ to adjust virtual link quality. ψ is calculated as *the percentage changes between current link quality and link quality of previous period*. When the motes surrounding a sender all experience link quality degradation or improvement, given the theoretical packet delivery success rate, P_{prr}, we have the adjusted packet reception rate P'_{prr} over virtual link as:

$$P'_{prr} = \psi \times P_{prr}. \tag{4}$$

By integrating the anisotropic and dynamic properties, MagicLink's adaptive radio model can emulate the radio communication in high-fidelity, and user applications' reliability and robustness can be thoroughly investigated.

4 Resource Mediation

Resource mediation is a collection of operations that maintains and updates sensor nodes and virtual link information. These operations mainly take place on gateway computers. Resource mediation functionalities corresponds to the initialization and monitoring components on the left hand side of virtual link pool in Figure 3. Initialization of testbed is a static, one-time operation, which includes user configuration and bootstrapping, while monitoring is a periodically executed procedure, which can be further categorized as sensor and Internet status monitoring. New status report message will trigger a virtual link records update in the virtual link pool. This update is essentially an adjustment of the theoretical link quality values in accordance to measured real link status. We elaborate these operations in this section and illustrate them in the order of temporal execution sequence within the lifetime of the testbed.

4.1 User Configuration

To setup MagicLink across multiple sites, as shown in Figure 1(b), users are asked to provide the following parameters through a user interaction interface to establish the initial platform topology:

- Information about how these SSNs to be connected, especially the gateways' network locations.
- Virtual link distances and environmental parameters, this information is used for initialize the adaptive radio connection models. The orthogonal distance of two adjacent SSNs, e.g., SSN 1 and SSN 2 in Figure 1(b), are provided by users to initialized the basic topology.
- Sensor status monitoring frequency, this parameter is adjustable depending on the specific user application features.

At this point, we assume that user has already obtained privilege of accessing these SSNs to construct their testbeds. We will not elaborate on the associated resource discovery and authentication mechanisms. The information provided by users will be used to evaluate the initial link qualities. By default, local SSN topology and transmission power information will be used.

4.2 Bootstrapping

Since users may not have all the sensor mote connection information, a bootstrapping phase is necessary to assist virtual link pool initialization. Based on the testbed federation information specified by a user, the procedure of bootstrapping includes: (1) identifying which locally connected sensor motes are selected as edge nodes for cross-site communication; (2) calculating the relative distances of edge nodes according to user defined orthogonal testbeds distance; (3) evaluating the default values for each link's quality metrics according to MagicLink's radio model. After bootstrapping, each sensor node in a local testbed will establish connections to several remote edge nodes as its neighbors in the federated platform.

4.3 Status Monitoring

The status monitoring component contains two parts running on different hardware: the *sensor network status monitor* running on each edge node and *Internet status monitor* running on each gateway computer.

Sensor network status monitor: In order to facsimile the important properties of radio links, Sensor network Status Monitor (SSM) is employed to assist virtual link simulation. The packet reception rate metric P_{prr} is of primary concern, whereas average transmission latency τ_s is protocol related and can be helpful in some cases, such as real-time communication [20]. SSM runs periodically and measures the aforementioned metrics to provide a reference for simulating physical channel characteristics.

One advantage of designing SSM as a configurable component is that it can easily be adjusted in accordance to different application contexts. First, depending on various usage scenarios, users can choose to enable or disable SSM at any time. Take a data gathering application for example, when the primary concern of the user application is to collect sensed data, rather than to investigate the network protocol behavior, SSM is unnecessary and can be safely turned off. On the other hand, if user application requires link status monitoring, SSM can either be reused by the user application to alleviate programming burden, or simply be replaced by user's own monitoring program as long as the same status report message format is used. SSM can also be customized in terms of scheduled execution time and frequency in accordance to user needs.

Once the SSM component is configured, it measures packet delivery rate by sending out probing message during the "idle" time of an application. During each probing period, a fixed number of messages are sent out, and the packet delivery ratio of a link is estimated by dividing the number of received packets by the expected packet number. To elaborate, for each edge node n_i, packet reception rates between each of its neighbors are measured. Given P_{prr}^{ij} as the reception rate between n_i and n_j, the average reception rate for edge node n_i with m neighbors is: $\sum_{j=0}^{m} P_{prr}^{ij}/m$.

Although transmission latency is a relatively less significant factor in sensor networks comparing to packet reception ratio (many simulators simply ignore this factor), in a hybrid system like MagicLink, transmission latency can be useful to some extent. The measurement for latency τ_s is provided as an optional function, and is evaluated by subtracting the time when a packet is sent from the time when an acknowledgement is received by the sender. Assume τ_s^{ij} is the packet transmission latency between sensor node n_i and n_j. For edge node n_i with m neighbors, its average one-hop transmission latency is: $\sum_{j=0}^{m} \tau_s^{ij}/m$

Internet status monitor: Similar to SSM, Internet Status Monitor (ISM) measures Internet link qualities between gateway computers at a user configured frequency. Packet reception rate and packet transmission latency are also relevant metrics we use to quantify link qualities. Depending on the underlying protocol used, Internet connections can be very reliable, thus measuring Internet reception rate is primarily used for preventing extreme cases, such as loss of connection.

As to packet transmission delay, there is a rich literature on accurately measuring transmission delays between Internet host computers, e.g, [21, 22]. We adopted the algorithm proposed in [22]. For gateway computers within the same Internet domain, average transmission delay τ_i is usually within 10 milliseconds. This is about the same delay as typical one-hop radio transmission without using any MAC or other protocols. If the SSM measured radio transmission delay τ_s is significantly larger than τ_i, which implies possible heavy data traffic or large packet size, the message forwarding component of MagicLink will interfere accordingly. On the other hand, for cross-domain Internet connection, a longer delay may present. If this is the case, a notification to users will generate for proper settings of timeout thresholds, if applicable.

5 Adaptation and Dispatching

Virtual link adaptation and message dispatching is the core operation that actually achieves resource stitching among distributed sensor network sites. It establishes a virtualization layer that hides the underlying geographical and connection heterogeneity. The objectives of link adaptation and messaging dispatching are manifold: (1) providing joint messaging interface between radio links and Internet; (2) performing virtual link lookup and forwarding packets to appropriate virtual links; (3) emulating radio link behavior by adaptively adjust message dispatching operations on virtual links according to real radio link environments and conditions.

5.1 Send/Receive Interface

On the edge sensor nodes of each SSN, whenever a radio message is broadcasted, both the radio interface and serial/USB interface should be involved such that local and remote neighboring nodes can hear this transmission. A customized send/receive interface, which is implemented on top of TinyOS, is provided in MagicLink to handle this job. Users invoke this interface the same way as the built-in radio send and receive functions in nesC, with the same destination address format.

5.2 Message Forwarding

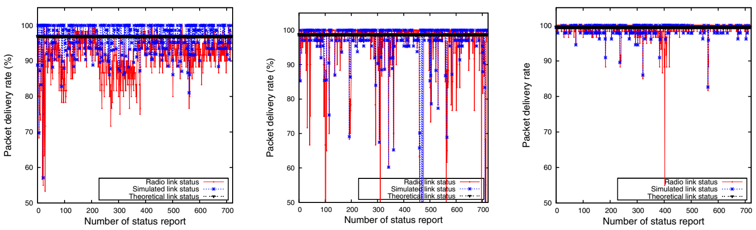
A message queue is implemented as a container to store sensor network application messages on a gateway computer. In addition to the raw messages transmitted within the sensor networks, time information is also included in each message stored in the message queue for referencing purpose by certain applications. Once an edge node transmission is heard by the gateway, which means the transmitted message should be forwarded onto the virtual links, the gateway will insert this message into the message queue. Messages are popped out and forwarded by the message dispatcher. Virtual link lookup operation is implemented to guarantee messages are forwarded to the proper destination. In addition, different virtual links may have different packet reception rates according to the adaptive ratio model explained in Section 3. Packet forwarder processes each message based on the adaptation rules and forwards it to the virtual links selected by the link lookup operation.

5.3 Adaptation

Virtual link adaptation refers to emulating the properties of wireless radio links on the Internet connections. Based on the virtual link quality measurements, some messages in the message queue may be intentionally dropped to simulate a packet loss. In other words, the time to forward packets on the virtual link is dependent on the adaptation policy configured by the end user in MagicLink. If a user turns on the transmission delay adaptation in MagicLink configuration, the packets forwarding time will be affected accordingly. Particularly, when the gateway connections delays are much less than radio transmission latency, the packet forwarding operation is intentionally postponed. If the latency on the Internet is similar to that on the radio links, packets are forwarded immediately.

6 Performance and Usage Cases

To validate our implementation of MagicLink system and demonstrate the effectiveness of the radio model used, we tested two representative usage scenarios, single hop communication and multihop communication, on a federated testbed constructed by MagicLink. The testbed is configured as follows: a total of 32 sensor nodes are deployed at University of Florida and Oklahoma State University. Each site configured 4×4 Telosb sensor motes arranged in a grid topology with a node to node distance of 9 feet. At each site, there is one gateway computer (fitPC2) connected to these sensor motes for reprogramming and power supply. Four motes from each site were configured as edge nodes. They all had SSM installed and communicated through the send and receive interface provided by MagicLink. Virtual links are set as 9 feet in distance as well, and use the same transmission power level as radio communications. The variance σ is set to 7.6 as adopted by NS-2 to indicate office environment with soft partitions.

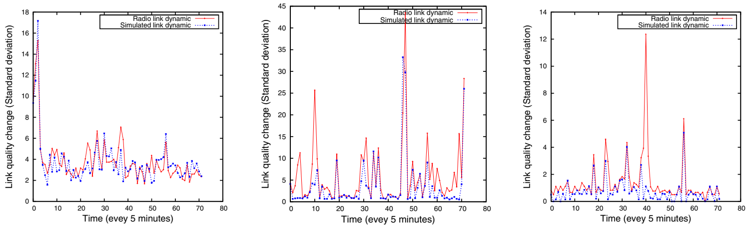


(a) Low transmission power level (b) Medium transmission power level (c) High transmission power level

Fig. 4. Single hop link quality comparison: real radio link quality versus virtual link quality

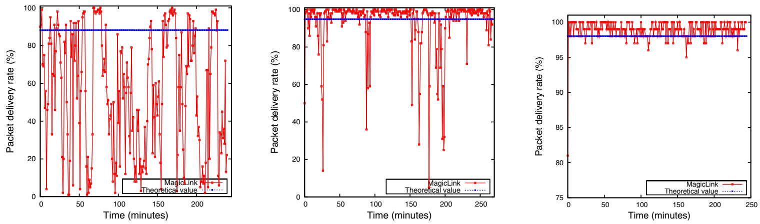
6.1 Single Hop Communication

One advantage of using MagicLink is to achieve smooth cross-site communication; we tested the performance of MagicLink's radio model by comparing the quality of real



(a) Dynamic changes during low power transmission (b) Dynamic changes during medium power transmission (c) Dynamic changes during high power transmission

Fig. 5. Single hop transmission dynamics: real radio link versus virtual link



(a) Low transmission power level (b) Medium transmission power level (c) High transmission power level

Fig. 6. Multihop communication across two sites at different transmission power levels

radio link with the virtual link. We randomly selected one edge node at each site to form a virtual link. Packet reception rate (PRR) is used as the comparison metric here. By letting one edge node send probe messages via the send interface at the frequency of 120 messages per second, both neighbors at the local site and the virtual neighbors at the remote site can hear this transmission. Upon receiving a message, an acknowledgement is transmitted back to the sender in order to collect status information. Three transmission power levels were tested, and virtual link adjustment was twice per second. This experiment was conducted continuously for 6 hours to thoroughly exploit the dynamic property of our radio model. We plot the results in Figure 4 and 5.

In these figures, the red lines show the fluctuation of radio link qualities, the straight blue lines stand for the theoretical packet reception rate, a constant value calculated by the methods used in NS-2 [23] and some other simulators, and the green lines represent the virtual link variations in MagicLink. From Figure 4 one can observe that the adaptive radio model used in MagicLink vividly emulates the lossy and dynamic behavior of radio transmissions, which is hard to manifest by the previously proposed theoretical radio simulation model in other projects. In addition, with the increase of transmission power, radio communication becomes more reliable, and both MagicLink and the constant model show a better result in approximating link quality. Figure 5 quantifies the dynamic changes of radio links by sampling the standard deviation of packet reception rate in every five minutes. Both radio link and virtual link exhibit similar fluctuation of link quality changes. At some rare cases, e.g., time 40 in Figure 5(c), one can observe that the standard deviations of virtual link and radio link are significantly different comparing to other approximations. This can be explained that the radio link dynamic created by the adaptation method in MagicLink is probabilistic in nature (refer to Equation 4), and the virtual link is also affected by the anisotropic property. Therefore, some cases may deviate greatly from the observed variations of the nearby real links. Similarly, this kind of difference appears also in two real radio links or two different time periods of the same radio link under identical environments, making our MagicLink virtual links behave like real radio links in these aspects too. However, MagicLink is designed to be flexible that with proper configurations, user can acquire the desired level of link emulation.

6.2 Multihop Communication

Multi-hop communication is one of the key features in many applications targeted at large-scale sensor networks. In this set of validation, we set up a multi-hop configuration implemented by MagicLink and recorded packet delivery rates at different transmission power levels. The implementation of a multi-hop configuration enabled by MagicLink is depicted as follows: from the federated testbed, we randomly selected one sensor node at each site to form a 4-hop communicate path, three of them are radio connections while the rest one is virtual link. We plotted the end-to-end packet delivery rate at different transmission power levels in Figure 6. From the figure we draw two conclusions. First, MagicLink successfully accomplishes cross-site communication. The pattern of the end-to-end packet transmission path agrees with many theoretical analysis, e.g., [24]. When transmission power level is low, end-to-end link quality is very poor and shows significant dynamics over time. With the increase of transmission power level, the overall link quality is improved. Again, the constant blue lines in all three figures show that the theoretical link simulation is not capable of capturing the dynamic changing link property. In contrast, the mixture multi-hop model represented by the red lines suggests that MagicLink seamlessly stitches the two simulation sites by rendering similar multi-hop communication patterns as within the same local sensor testbed.

7 Related Work

The maturing of sensor network technologies has resulted in increasing demand for large-scale sensor experiment platforms for fast prototyping and experimentation. Due to the high cost of sensor motes and computation/memory limitations for high-fidelity simulation, existing approaches for providing such experiment platform mainly fall into two categories: hybrid simulation and testbeds virtulization.

Hybrid simulation approach emphasizes on using real sensor network data, such as radio link status measurements and sensed data, as the input to simulators to improve simulation quality. This approach is adopted by SensorSim [26] and Kansei [25]. In SensorSim, real sensed data from a limited number of sensor motes are collected as input to the simulator in order to test applications' response to environmental changes. In Kansei, the sensor network simulator on a computer is connected to real sensor nodes, and the radio communication is performed by real radio hardware on these sensor nodes. Their methods are viable to small-scale simulations, but may not be applicable to large-scale simulations due to message congestion.

Testbeds virtualization, on the other hand, focuses on creating a mapping between deployed real testbeds and virtual nodes in a computer to visualize testbeds topology and communication paths. Typically, in a virtualized testbed, there are more simulated sensor nodes than real sensor nodes. For example, in NetTopo [7] the authors created avatars for each real sensor node in a simulator. The WISEBED project [8] and [6] proposed a testbed federation method that is conceptually similar to MagicLink's design. However, despite the conceptual similarity, a high-fidelity radio model that is critical for seamless cross-site communication is missing from all these existing approaches, and MagicLink is the essential missing piece for accomplishing smooth testbeds federation and virtualization.

An essential component for sensor network experiment platform is its radio communication module. To better reflect the communication performances, there exist a rich literature on modeling of radio link properties. Two broad classes for modeling are widely used in simulation of radio transmission behaviors in WSNs, namely analytical models and empirical data based models.

For analytical modeling of radio signal attenuation, the physical layer features of a wireless link are captured by the radio signal attenuation formulas. It provides a simplified and straightforward description of wireless communication between sensor nodes. Many simulators adopt this approach to build up connections in a simulated network. NS-2 [23] calculates received power at a receiver according to a user defined distance. This power level is compared to (1) the receive threshold (RT), and (2) the carrier sense threshold (CST). If it falls below CST, the packet is discarded as noise. If the received power is between CST and RT, the packet is marked as an error packet. Otherwise, if received power is above RT, the packet is conceived normal. Similar approaches that using theoretical model to determine packet transmission rate are also adopted by GloMoSim [27], ATEMU [28], and SWAN [29]. In MagicLink, we leverage the analytical models to setup initial values for each virtual link. However, to reflect dynamic features of radio communications, MagicLink periodically adjusts virtual link quality metrics and values at runtime. This continuous refreshing process is a great leap forward towards more realistic radio model, and makes Magiclink suitable for investigating the reliability and robustness of real-world sensor applications.

An alternative approach to model radio links is to use empirical data. With the help of radio quality trace files, simulators can provide diverse environment settings for radio link simulations, making it a more favorable approach to many simulator implementations. In TOSSIM [30], the simulator loads the empirical data files to generate statistical models for each link. Although this approach offers more flexibility, it consumes huge memory space for simulating a network (e.g., 10MB per node), which hinders its application to large-scale network simulation. Several researches used smoothing and fitting methods to statistically analyze the experimental data samples, e.g., [31] and [32]. Instead of using the one-time-gathered trace data, MagicLink's radio link model includes an online algorithm that is capable of dynamically adjusting virtual link status based on the monitored real links. since anisotropic property is also important for building a convincing radio link model. Zhou et al. [18] proposed a radio irregularity parameter, Degree of Irregularity (DOI), to quantify radio propagation patterns in sensor networks. MagicLink incorporates this DOI metric and assimilates the anisotropic feature to build a topology-aware virtual link for comprehensive radio transmission simulation.

8 Conclusion and Future Work

We proposed the MagicLink middleware system for building large-scale sensor network testbeds from distributed small sensor networks. MagicLink features an elastic infrastructure that flexibly integrates multiple sites of sensor networks. In order to help investigation of network protocol performance, we further proposed an adaptive radio communication model that embodies lossy, dynamic, and irregular properties of radio links. We experimentally tested representative sensor network applications on a testbed constructed by MagicLink, and showed that MagicLink possesses the desired features and reflects real radio dynamics and heterogeneity. Under the GENI initiative, using MagicLink, we are building a federated large-scale sensor network testbed that integrates multiple sensor networks across the nation with friendly web interfaces, visualization and debugging utilities. This large-scale testbed with high-fidelity "magic" virtual links is an ideal platform for evaluating and testing next-generation Internet protocols that consider sensor networks and mobile devices as first-class citizens.

References

1. TinyOS, `http://docs.tinyos.net/index.php/TOSSIM`
2. Werner-Allen, G., Swieskowski, P., Welsh, M.: Motelab: A wireless sensor network testbed. In: IPSN, pp. 483–488. IEEE (2005)
3. MistLab, `http://mistlab.csail.mit.edu/`
4. Sridharan, M., Zeng, W., Leal, W., Ju, X., Ramnath, R., Zhang, H., Arora, A.: From Kansei to KanseiGenie: Architecture of Federated, Programmable Wireless Sensor Fabrics. In: Magedanz, T., Gavras, A., Thanh, N.H., Chase, J.S. (eds.) TridentCom 2010. LNICST, vol. 46, pp. 155–165. Springer, Heidelberg (2011)
5. Sakamuri, D.: NetEye: A wireless sensor network testbed, Ph.D. dissertation, Wayne State University (2008)
6. Baumgartner, T., Chatzigiannakis, I., Danckwardt, M., Koninis, C., Kröller, A., Mylonas, G., Pfisterer, D., Porter, B.: Virtualising Testbeds to Support Large-Scale Reconfigurable Experimental Facilities. In: Silva, J.S., Krishnamachari, B., Boavida, F. (eds.) EWSN 2010. LNCS, vol. 5970, pp. 210–223. Springer, Heidelberg (2010)
7. Shu, L., Wu, C., Zhang, Y., Chen, J., Wang, L., Hauswirth, M.: NetTopo: beyond simulator and visualizer for wireless sensor networks. ACM SIGBED Review (2008)
8. The WISEBED project web page, `http://www.wisebed.eu/`
9. Foster, I., Iamnitchi, A.: On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, pp. 118–128. Springer, Heidelberg (2003)

10. Michael, A., Armando, F., Rean, G., Anthony, D., Randy, K., Andy, K., Gunho, L., David, P., Ariel, R., Ion, S., et al.: Above the clouds: A berkeley view of cloud computing. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28 (2009)
11. Li, Z., Wang, N., Franzen, A., Li, X.: Development of a wireless sensor network for field soil moisture monitoring. In: ASABE (2008)
12. Liu, X., Zhao, H., Yang, X., Li, X., Wang, N.: Trailing Mobile Sinks: A Proactive Data Reporting Protocol for Wireless Sensor Networks. In: Proc. of MASS (2010)
13. Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., Anderson, J.: Wireless sensor networks for habitat monitoring. In: Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, pp. 88–97. ACM (2002)
14. Goldsmith, A.: Wireless communications. Cambridge Univ. Pr. (2005)
15. Chipara, O., Hackmann, G., Lu, C., Smart, W., Roman, G.: Practical modeling and prediction of radio coverage of indoor sensor networks. In: Proc. of the ACM/IEEE International Conference on Information Processing in Sensor Networks (2010)
16. Stoyanova, T., Kerasiotis, F., Prayati, A., Papadopoulos, G.: Evaluation of impact factors on RSS accuracy for localization and tracking applications. In: Proc. of the 5th ACM International Workshop on Mobility Management and Wireless Access, pp. 9–16. ACM (2007)
17. Zamalloa, M., Krishnamachari, B.: An analysis of unreliability and asymmetry in low-power wireless links. ACM Transactions on Sensor Networks, TOSN (2007)
18. Zhou, G., He, T., Krishnamurthy, S., Stankovic, J.: Models and solutions for radio irregularity in wireless sensor networks. ACM Transactions on Sensor Networks, TOSN (2006)
19. CC2420 radio datasheet 2nd ed. Texas Instruments (October 2005)
20. He, T., Stankovic, J., Lu, C., Abdelzaher, T.: SPEED: A stateless protocol for real-time communication in sensor networks. In: Proc. of ICDCS (2003)
21. The IDMaps web page, `http://idmaps.eecs.umich.edu/`
22. Gummadi, K., Saroiu, S., Gribble, S.: King: Estimating latency between arbitrary internet end hosts. In: Proc. of the 2nd ACM SIGCOMM Workshop on Internet Measurment. ACM (2002)
23. NS-2, `http://www.isi.edu/nsnam/ns/`
24. Zhao, J., Govindan, R.: Understanding packet delivery performance in dense wireless sensor networks. In: Proc. of SenSys. ACM (2003)
25. Ertin, E., Arora, A., Ramnath, R., Nesterenko, M., Naik, V., Bapat, S., Kulathumani, V., Sridharan, M., Zhang, H., Cao, H.: Kansei: a testbed for sensing at scale. In: IPSN. IEEE (2006)
26. Park, S., Savvides, A., Srivastava, M.: SensorSim: a simulation framework for sensor networks. In: Proc. of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems. ACM (2000)
27. Ben Hamida, E., Chelius, G., Gorce, J.M.: Impact of the physical layer modeling on the accuracy and scalability of wireless network simulation. Simulation (2009)
28. Polley, J., Blazakis, D., McGee, J., Rusk, D., Baras, J.: Atemu: A fine-grained sensor network simulator. In: IEEE SECON (2004)
29. Liu, J., Perrone, L., Nicol, D., Liljenstam, M., Elliott, C., Pearson, D.: Simulation modeling of large-scale ad-hoc sensor networks. In: Proc. of European Interoperability Workshop (2001)
30. Levis, P., Lee, N., Welsh, M., Culler, D.: TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In: Proc. of SenSys. ACM (2003)
31. Cerpa, A., Wong, J., Kuang, L., Potkonjak, M., Estrin, D.: Statistical model of lossy links in wireless sensor networks. In: IPSN (2005)
32. Sundresh, S., Kim, W., Agha, G.: SENS: A sensor, environment and network simulator. In: Proc. of the 37th annual symposium on Simulation. IEEE Computer Society (2004)