Rapidly-Deployable Mesh Network Testbed

Michael R. Souryal, Andreas Wapf, and Nader Moayeri
National Institute of Standards and Technology
Gaithersburg, Maryland, USA
souryal@nist.gov, andreas@wapf.ch, moayeri@nist.gov

***Abstract*—This paper describes a wireless mesh network testbed for research in rapid deployment and auto-configuration of mesh nodes. Motivated by the needs of first responders and military personnel arriving to an incident area, we developed and tested an automated deployment algorithm that indicates when a mesh node needs to be deployed as the coverage area grows. Conventional radios can experience severe coverage limitations inside structures such as hi-rise buildings, subterranean buildings, caves, and underground mines. The approach examined here is to deploy wireless relays that extend coverage through multihop communication using a deployment algorithm that employs physical layer measurements. A flexible platform based on IEEE 802.11 radios has been implemented and tested in a subterranean laboratory complex where conventional public safety radios have no coverage. Applications tested include two-way voice, data, and location information. This paper describes the testbed, presents experimental results, and recommends areas for further study and development in rapidly-deployable multihop networks.**

I. Introduction

In certain applications such as emergency response there is a need for reliable, rapidly-deployed communications to and among users in an incident area. Unfortunately, conventional radio communications technologies can leave coverage gaps, particularly inside large buildings and in subterranean structures where radio signals can be severely attenuated, resulting in disconnected users.

This paper investigates an approach to enhance coverage in such problematic environments using rapidly deployed wireless relays. The relays form a wireless mesh network backbone that serves to connect users to each other. A key objective here is to automate the deployment process so as to minimize the impact on a user's mission. Each mobile user would carry a cache of small, inexpensive wireless relays (e.g., Fig. 1), also referred to as *breadcrumbs* [1], and would deploy a relay when instructed to do so by the system. In some scenarios, the relays may be deployed by robots remotely controlled to search an incident area. Due to the time-sensitive and critical nature of applications such as emergency response, the backbone must be deployed rapidly and yet support highly reliable communications services, two often contradictory objectives.

In previous work, Bao and Lee [2] proposed a collaborative deployment protocol to reduce redundant relay deployment and examined its performance through simulation. Refaei *et. al.* [3] used simulations to examine the use of cognitive radios for interference mitigation in rapidly deployed networks.

In experimental work with prototypes, Naudts *et. al.* [4] developed a network monitoring and planning tool which assisted in node deployment. Pezeshkian *et. al.* [5] developed

Fig. 1. Relay implementation (external view)

and tested an automatic deployment system for robotic applications. Both of these systems were based on IEEE 802.11 and used received signal strength indication (RSSI) measurements to trigger the deployment of new nodes. In similar work, we developed a prototype based on 900 MHz TinyOS motes which also used RSSI for deployment [6].

In other related work, researchers at Virginia Tech, in collaboration with SAIC, investigated rapidly-deployable wireless communications for the wider incident area (on the order of several kilometers) using, for example, Local Multipoint Distribution Service (LMDS) [7]. Similar to the breadcrumb concept, but with the added capability of autonomous mobility for fine-tuning the relay's location, the Defense Advanced Research Projects Agency initiated the *LANdroids* program to solve communications problems faced by warfighters in urban environments [8].

The study of this paper differs from previous work in several respects. First, it extends our previous simulation results [3] with experimental results obtained from a testbed implementation. Second, the deployment algorithm uses bi-directional link measurements of signal-to-noise ratio (SNR) to take into account the potential asymmetry of each link. Third, this study presents a more detailed and quantitative performance analysis than other published studies using such metrics as packet delay, throughput, and packet loss, both during and after deployment of the network, and encompassing interactive two-way voice as well as data applications.

Section II reviews link measurement studies that were used to guide the design of our deployment algorithm. Section III provides details of our testbed implementation. In Section IV, we present results of experiments in a subterranean laboratory

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

complex that has no coverage from ground-based radio services and currently represents a dead-zone for public safety personnel. Our experiments demonstrate deployment of a 9-hop network with round-trip delay under 80 ms. Conclusions are presented in Section V along with suggestions for future work.

II. Link Measurements

An efficient relay deployment protocol requires an ability to measure wireless link quality in order to determine when a new relay needs to be deployed. One useful definition of link quality in a store-and-forward type of network is the reliability of the link, or the probability that a message transmitted on the link is successfully received. Other possible definitions could be related to the throughput or delay of the link, but most will depend at least in part on link reliability.

In previous work, we experimentally investigated various approaches for measuring link reliability, including packet counting and mapping signal-to-noise ratio (SNR) measurements to packet success rate, on both static [9] and time-varying [10] links. In the latter work, we examined the performance of various filters for the raw SNR measurements provided by the physical layer, including a simple moving average, an exponential moving average, and a Yule-Walker predictor. Using IEEE 802.11b/g radios at various data rates as a case study, we concluded that the measured, filtered SNR from commercially available devices is predictive of link reliability. We note that this conclusion stands in contrast to that of [11], for example, in which little correlation between the two was observed. In [9], we identified conditions which adversely affect the correlation.

In [10], we also observed that link quality assessment using SNR exhibited lower latency than using packet counting on time-varying links. The latency issue is critical for real-time deployment, especially considering that link reliability was observed to drop rapidly around a certain SNR. If a deployment algorithm were to rely on packet counting to determine when to deploy a relay, by the time packet losses are detected on a mobile link, it may very well be too late to deploy a relay. Physical layer measurements such as SNR permit early detection of potential marginal links before they occur. Furthermore, knowledge of the current SNR of a time-varying link allows one to build a fade margin into the link budget for a deployed link.

III. Testbed Design

Drawing on the observations of the aforementioned studies, this section describes the design of our rapidly-deployable relay. We review our hardware and software design choices and describe the deployment algorithm which indicates to the mobile user when a relay needs to be deployed.[1]

[1]Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

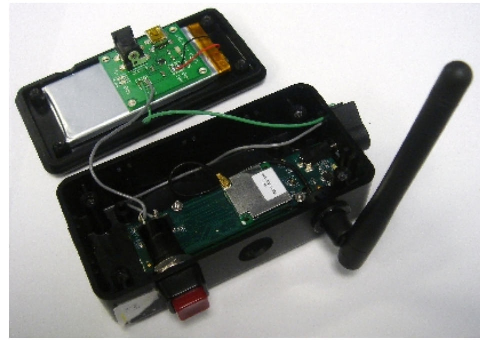


Fig. 2. Relay implementation (internal view)

A. System Details

Each relay consists of a miniature Linux computer, IEEE 802.11b/g (WiFi) radio, antenna, rechargeable battery, and power switch, contained in a $10.2\,\mathrm{cm} \times 5.1\,\mathrm{cm} \times 2.5\,\mathrm{cm}$ ($4\,\mathrm{in} \times 2\,\mathrm{in} \times 1\,\mathrm{in}$) plastic enclosure (Fig. 2). The miniature computer (Gumstix connex 400xm) contains a 400 MHz Intel XScale® PXA255 processor, 16 MB flash memory, and 64 MB SDRAM, and runs a Linux 2.6 kernel.

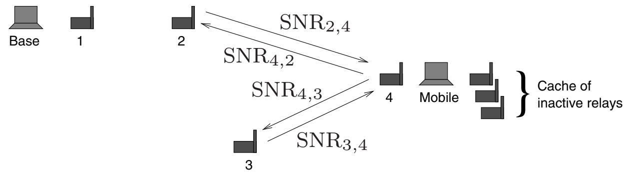
An expansion board (Gumstix wifistix) provides the WiFi capability using the Marvell® 88W8385 chip. The source code of this module's driver is available under a General Public License, and we modified it to pass the signal and noise measurements of each received packet to the upper layers. This functionality enables an application to monitor the quality of a link, as required for the deployment algorithm described below. The WiFi expansion board comes with a 2 dBi dipole antenna.

A rechargeable lithium polymer battery (Ultralife 3.7V 930mAh) provides power for one to two hours of active use. An integrated lithium polymer battery charger allows charging of the battery in place.

B. Routing Protocol

While the Linux operating system provides the relay with a standard IP stack including TCP and UDP transport layers, a routing protocol is needed for the dynamically configured and changing wireless mesh network. We chose to use Optimized Link State Routing (OLSR), a proactive routing protocol, because we anticipate most of the deployed network to be fixed with route changes occurring only near the mobile users. Furthermore, for the intended application of indoor emergency response, scalability beyond tens of nodes is not a major concern. We selected a user space implementation of OLSR [12] and used its optional link quality extension which evaluates routes with the Expected Transmission Count (ETX) link metric [13]. This metric takes into account the reliability of each link on a path rather than just its number of hops, resulting in higher quality routes.

Even though the 802.11b/g radio is capable of unicasting up to 54 Mbps, we configured it to make all transmissions using the 2 Mbps data rate. The reason behind this design choice is that the ETX link metric is calculated based on an estimate of



$\mathrm{SNR}_2 = \min\{\mathrm{SNR}_{2,4}, \mathrm{SNR}_{4,2}\}$
$\mathrm{SNR}_3 = \min\{\mathrm{SNR}_{3,4}, \mathrm{SNR}_{4,3}\}$
Deploy relay 4 if:
$\max\{\mathrm{SNR}_2, \mathrm{SNR}_3\} < T_{\mathrm{deploy}}$

Fig. 3. Deployment illustration in which mobile node 4 probes and deployed relays 2 and 3 reply; $\mathrm{SNR}_{i,j}$ is the filtered SNR of link i-j.

packet success rate obtained by counting HELLO messages which are broadcast at the mandated rate of 2 Mbps. Since higher data rates require higher quality links, there would otherwise be a mismatch between the quality of the links on the selected routes and the quality required by higher data rates. We propose below changes to the link metric which would better support higher data rates.

C. Deployment Algorithm

The deployment algorithm monitors the quality of a mobile node's links to the network and tells the user when a new relay needs to be deployed.[2] The protocol is described as follows, and an example is illustrated in Fig. 3. A mobile node continuously broadcasts probe packets (at a rate of 10 probes per second). Previously deployed relays in the network are configured to listen for and reply to the probes. Specifically, upon receipt of a probe message, a receiving relay records the SNR measured on receipt of the probe and embeds this measurement in a reply message. Between probe broadcasts, the mobile node listens for replies and records the SNR of each received reply. In this way, the mobile node obtains bidirectional SNR measurements of each detected link, and the deployment algorithm is able to factor in the potential asymmetry of a link. The built-in carrier sense multiple access scheme with collision avoidance (CSMA/CA) of 802.11 decreases, though does not eliminate, the probability of collision among the replies.

When a node receives probes from more than one mobile node, one efficiency would be to aggregate multiple SNR measurements into a probe reply. With four probing nodes and four replying nodes all within range of one another, for example, aggregation would reduce the average channel utilization of the probe requests and replies from $13\,\%$ to $5\,\%$. Another efficiency is to adapt the probe rate to the mobility of the probing node, which we investigated in [6].

For each detected link, the mobile node filters the raw SNR measurements in each direction. Currently, this filter is a simple moving average over a fixed window of the last 20 probe periods. The *link SNR* is defined as the smaller of the two unidirectional SNR estimates. When the highest of all the link SNRs falls below a threshold, T_{deploy}, an indication is given to deploy a node. If the highest SNR falls below a lower threshold, T_{critical}, a warning is issued. In our implementation, T_{deploy} is set to 25 dB and T_{critical} is set to 20 dB, which still leaves about a 15 dB margin above the DS/CDMA chip SNR at which link reliability drops precipitously.

This algorithm waits to trigger deployment until the strongest link falls below the threshold. To provide link diversity and redundancy, the algorithm could easily be modified to trigger deployment when the L^{th} strongest link falls below the threshold, providing L-order link diversity against fading.

The deployment indication is given by the probing node in the form of a remote procedure call issued to a server application on a tablet PC carried by the mobile user. The server application displays one of three messages depending on whether the highest link SNR is above T_{deploy} (good link), between T_{deploy} and T_{critical} (deploy node), or below T_{critical} (critical).

IV. Experimental Results

We tested the system in a subterranean laboratory complex on the campus of the National Institute of Standards and Technology, Gaithersburg, Maryland, USA. Buildings 218 and 219, part of the Advanced Measurement Laboratory, consist of two underground levels, a basement level approximately 7 m under ground and a sub-basement level approximately 11 m under ground (Fig. 4).

A. Deployment Example

Deployments consisted of a network of relays connecting two end stations, a base unit and a mobile unit. While the system is not inherently limited to two end stations, the purpose of the tests was to examine the feasibility and reliability of rapid deployment rather than measure system capacity in number of users.

A total of nine relays was used in addition to a tablet PC representing the mobile unit and a laptop representing the base unit. Both the tablet and the laptop had a WiFi interface and ran a version of the Linux operating system as well as the OLSR protocol. Four different deployment trials were conducted over a two-day period with similar resulting topologies. Measurements were collected during the 1 h to 2 h period afforded each trial by the batteries in use. Presented below are the details of one such trial during which the most extensive measurements were made.

The base laptop was placed in a semi-open room on the ground level of Building 217, across the hall from a stairwell that leads to the sub-basement level of Building 219. All nine relays were turned on and started in a default inactive state. The first relay was activated as a router and placed next to the base. The second relay was activated as a probing router. The purpose of placing relay 1 next to the base laptop was to reply to the probes transmitted by relay 2. Relays 2 through 9 and the tablet PC were then carried by the mobile user through the metal door to the stairwell.

[2]This indication could ultimately be integrated in an automated relay deployment system mounted on the user or on a robot, as in [5], but in our testbed it simply indicates to the user to manually activate and deploy a relay.

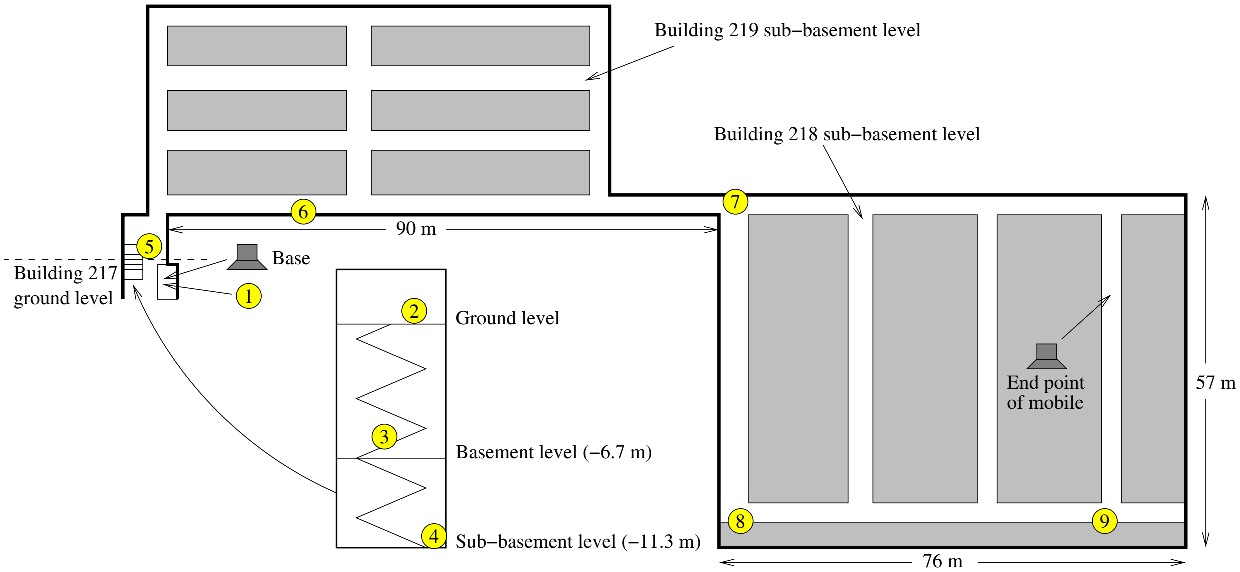


Fig. 4. Maps of sub-basement level of Buildings 218 and 219 on the NIST Gaithersburg campus

When (the probing) relay 2 indicated the need to deploy a node, relay 2 was placed on the floor. Probing was turned off on relay 2 and activated on relay 3. At this point, relay 2, like relay 1, routed data packets and replied to relay 3's probes. The user then continued along a path away from the base. The process of activating a relay to probe, continuing along a path away from the base, and deploying it upon indication to do so by the deployment algorithm was repeated until all nine relays were deployed. The objective was to stretch the network as much as possible, resulting in a logical topology that tended to be linear.

Fig. 4 shows where the relays were deployed in the stairwell descending to the sub-basement level of Building 219 and along the sub-basement corridors of Buildings 218 and 219. Measurements of throughput, delay, and packet loss were collected both during and after deployment for subsequent analysis and are described below.

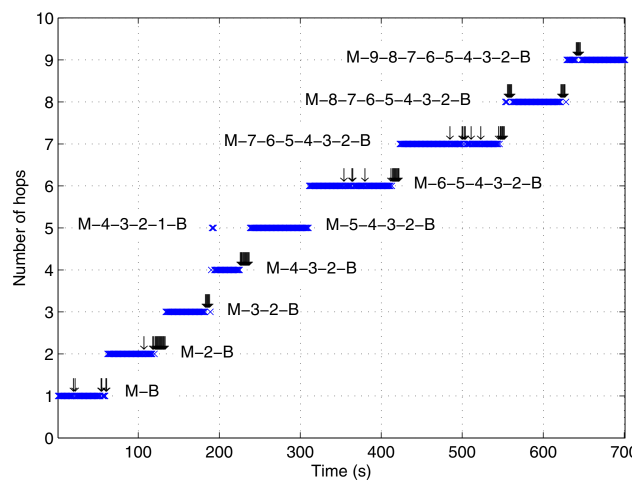


Fig. 5. Number of hops vs. time during deployment (×); down arrows (↓) indicate packet losses; text labels indicate routes from mobile (M) to base (B)

B. Performance During Deployment

1) Number of Hops and Round-trip Packet Loss: At the outset of the trial, a full-duplex voice-over-IP (VoIP) call was initiated between the mobile tablet PC and the stationary laptop. The audio-only call was made using a Linux version of Linphone, an open-source session initiation protocol (SIP) client. The voice codec used was 16 kHz speex at 28 kbps. The call was active during the entire deployment phase.

Simultaneously with the voice call, two end-to-end ping sessions were initiated, one from each end station. The purpose of the ping sessions was to monitor round-trip delay, packet loss, and the route taken by the ping packets between the mobile user and the base laptop as the user moved away from the base and relays were being deployed. Fig. 5 plots the one-way number of hops traversed by each ping packet as indicated by the time-to-live field in the IP header. The downward pointing arrows represent packet losses as indicated by a gap in the ICMP sequence numbers. The text labels indicate the route between the mobile (M) and base (B). For brevity we only show results of the ping session initiated by the mobile; similar results are obtained from the ping session initiated by the base.

The results in Fig. 5 show the gradual increase in hops as the mobile moves away from the base and deploys relays to maintain connectivity. Most of the time, packets to and from the base were routed directly through relay 2, bypassing relay 1 which was adjacent to the base. The short-lived increase from 4 to 5 hops at $t = 191$ s was due to the temporary use of relay 1 as a router between the base and relay 2. Overall, 12 % of the round-trip ping packets were lost during deployment. We observe that most of the packet losses occurred during route changes, an observation we also

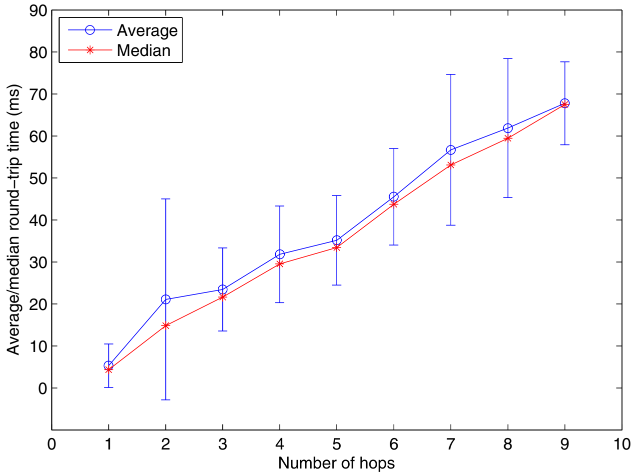


Fig. 6. Round-trip time by number of hops

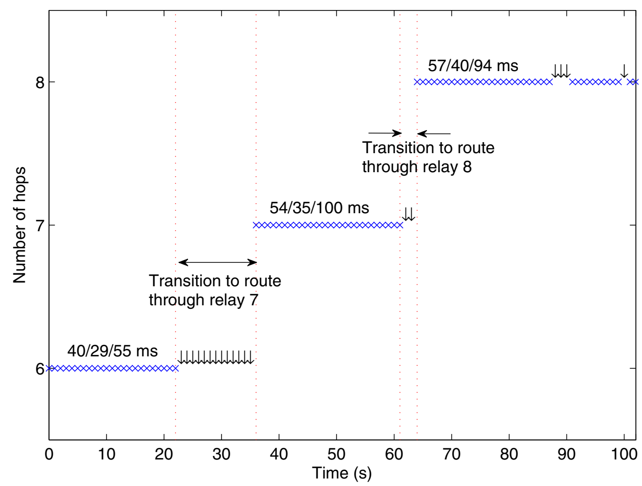


Fig. 7. Number of hops vs. time during audio test (×); down arrows (↓) indicate packet losses; text labels indicate avg/min/max round-trip times during each hop count

made during the audio test described below. Later, we propose improvements to the routing protocol to reduce these losses.

2) Round-trip Delay: Fig. 6 plots the average and median round-trip time as a function of the number of hops. The error bars represent one standard deviation above and below the mean. We observe that the relationship between round-trip time and number of hops is close to linear, with an average round-trip delay of just under 8 ms per hop, or 4 ms per hop one-way. Considering that subjective analysis has shown one-way voice delays of up to 150 ms to be hardly noticeable [14], significantly longer hop counts than those tested here may presumably be feasible for two-way voice communications. Such extrapolations must be tempered by the fact that these tests only comprised one source-destination pair, and that additional traffic sources would eventually introduce delays due to congestion. On the other hand, as noted earlier all transmissions in these tests were limited to 2 Mbps, and the use of higher data rates could increase network capacity.

C. Post-Deployment Performance

1) File Transfer: After deploying all nine relays, the mobile user stopped at the location indicated in Fig. 4 in the sub-basement level of Building 218. The two-way VoIP call was terminated as were both ping sessions. Using the secure copy client, scp, a 10 MB data file was transferred from the mobile user's tablet below ground to the stationary laptop on the surface over the 9-hop route M-9-8-7-6-5-4-3-2-B. The transfer completed in 8 min 3 s for an average throughput of 166 kbps. The client reported a peak throughput of 232 kbps. Prior to launching the file transfer, a new ping session was initiated from the mobile to the base which ran concurrently with the file transfer. This ping session revealed that the route remained as stated above for all but 6 s of the transfer during which it increased to 10 hops when it temporarily routed through relay 1. The higher packet loss rate and round-trip time of this ping session, 35 % and 171 ms, respectively, reflect the congestion generated by the file transfer. The file transfer throughput measurement indicates that the end-to-end capacity of reliable transfer over the 9-hop network is about 10 % of an individual link's raw physical layer data rate (2 Mbps).

2) Audio Performance with Route Changes: In addition to the file transfer, an audio test was performed in which a VoIP call was set up between the laptop and tablet. Linphone was used again with the same codec as before (16 kHz speex at 28 kbps). An audio narration of the Gettysburg Address [15] was played on loop into the input of the tablet, and the real-time decoded output was recorded at the laptop.[3] This audio experiment was conducted while the mobile user carried the tablet along the path of the already deployed relays in order to observe the effects of route changes on the voice connection. Concurrent with the audio were two end-to-end ping sessions, one from each end station, which were used to monitor the routes and delays.

Fig. 7, based on the ping session from the base to the mobile, shows the number of hops and packet losses for one cycle of the 102 s speech which spanned two route transitions. Also noted are the average, minimum, and maximum round-trip delays (avg/min/max) during each hop count. The speech began while the mobile user was walking between relays 6 and 7 away from the base, with the 6-hop route from the base to the mobile being B-2-3-4-5-6-M. The audio quality was reasonably good for most of the speech. Between $t = 22$ s and $t = 35$ s, the audio became choppy and intermittent as the mobile user passed relay 7 and the route was updated to 7 hops (B-2-3-4-5-6-7-M). After the route was updated, the audio quality was good again. At $t = 60$ s, the next route transition began as the mobile passed relay 8, and the route was updated to 8 hops (B-2-3-4-5-6-7-8-M). Here, the audio was choppy for only a few seconds. Around $t = 88$ s, there was an unexplained gap of about 4 s. The quality was good until the last few seconds of the speech during which the audio became choppy again as the route began to transition to 9 hops.

[3]The recorded output is available online at http://www.antd.nist.gov/~souryal/audio/gettysburg_address_received.wma

The packet losses and degraded audio quality around the route transitions can be explained from an understanding of how the routing protocol makes its route selections. OLSR with the ETX metric evaluates routes based on a crude estimate of the packet success probability of each link. A node obtains this estimate by counting the number of HELLO messages it receives from each of its neighbors in a recent window of time. In our configuration of OLSR, HELLO packets are broadcast at half-second intervals, and the link quality window size is 20 messages. This approach to packet success rate estimation has an inherent latency on the order of the window duration, or 10 s in this case.

As the mobile passed relay 7 towards relay 8, it rounded a corner, generating a precipitous drop in SNR. It took many seconds for the routing protocol to detect that the new route B-2-3-4-5-6-7-M was "better" (i.e., resulted in a lower cumulative ETX) than the current route B-2-3-4-5-6-M. The same thing occurred when the mobile passed relay 8 and transitioned from route B-2-3-4-5-6-7-M to B-2-3-4-5-6-7-8-M. The reason the audio degradation lasted longer during the first transition from 6 to 7 hops and was shorter during the second transition from 7 to 8 hops (13 s versus 3 s) can be deduced from the topology. The distance between relays 6 and 7 is longer than the distance between relays 7 and 8 (see Fig. 4), meaning that when the mobile rounded the corner near relay 7, it had a lower link budget from which to draw to maintain the link 6-M before the route was updated to go through relay 7. By contrast, when rounding the corner near relay 8, the shorter link afforded the routing algorithm more time to update the route to utilize relay 8 before link 7-M was broken.

The preceding observations point to a need for the routing algorithm to evaluate link quality with reduced latency. Instead of relying on a count of HELLO messages to estimate link quality, the routing protocol can utilize other measures, like SNR, to more rapidly and proactively respond to changing link quality. We expect that applying the same methods for link quality estimation used in our deployment algorithm to the routing protocol would substantially reduce the breaks in connectivity during route transitions. In addition to being able to better anticipate needed route changes, an SNR-based link quality metric would be better suited to support higher data rates. Using mappings of SNR to packet success rate that are specific to each data rate [10], the ETX (or other link quality metric) can be evaluated for the data rate of choice. Such changes to the route metric computation and their performance evaluation are subjects of future work.

V. Summary and Conclusion

A rapidly-deployable wireless mesh network is of value to emergency personnel, military personnel, or other users who need to establish communications in hard-to-reach areas in a time sensitive fashion. Because of the demanding and critical nature of such missions, it is desirable that the deployment process be automated to the extent possible, and yet yield highly reliable communication paths. To this end, we devised an automated deployment algorithm and implemented it on a flexible testbed platform of IEEE 802.11-based radios. The deployment mechanism utilizes bi-directional SNR link measurements to proactively determine when a node needs to be added to the network. We tested the rapidly-deployable multihop system in a subterranean complex where conventional public safety radio service is unavailable. The results of several trials revealed that the proposed deployment mechanism does yield well-connected multihop networks, up to 9 hops in our tests, with sufficiently low delay to support two-way voice. The experiments also revealed areas of needed improvement in the mesh routing protocol, such as making route transitions with lower latency. We proposed applying our techniques for link quality evaluation to the route metric computation in order to both lower the latency of route updates as well as to better support higher data rates.

References

[1] "BreadCrumb(R) wireless network user guide for the BreadCrumb(R) wireless network release 9.0," Rajant Corporation, Aug. 18, 2006.

[2] J. Q. Bao and W. C. Lee, "Rapid deployment of wireless ad hoc backbone networks for public safety incident management," in *Proc. IEEE GLOBECOM*, Nov. 2007, pp. 1217–1221.

[3] M. T. Refaei, M. R. Souryal, and N. Moayeri, "Interference avoidance in rapidly deployed wireless ad hoc incident area networks," in *Proc. 2nd IEEE Workshop on Mission-Critical Networking (in conjunction with IEEE INFOCOM'2008)*, Apr. 2008.

[4] D. Naudts, S. Bouckaert, J. Bergs, A. Schoutteet, C. Blondia, I. Moerman, and P. Demeester, "A wireless mesh monitoring and planning tool for emergency services," in *Proc. Workshop on End-to-End Monitoring Techniques and Services (E2EMON'07)*, May 2007.

[5] N. Pezeshkian, H. G. Nguyen, and A. Burmeister, "Unmanned ground vehicle radio relay deployment system for non-line-of-sight operations," in *Proc. 13th IASTED International Conference on Robotics & Applications*, Aug. 2007.

[6] M. R. Souryal, J. Geissbuehler, L. E. Miller, and N. Moayeri, "Real-time deployment of multihop relays for range extension," in *Proc. ACM MobiSys*, Jun. 2007, pp. 85–98.

[7] S. F. Midkiff and C. W. Bostian, "Rapidly-deployable broadband wireless networks for disaster and emergency response," in *Proc. First IEEE Workshop on Disaster Recover Networks (DIREN '02)*, Jun. 2002.

[8] "LANdroids broad agency announcement (BAA)," Information Processing Technology Office (IPTO), Defense Advanced Research Projects Agency (DARPA), BAA 07-46, June 5, 2007.

[9] M. R. Souryal, L. Klein-Berndt, L. E. Miller, and N. Moayeri, "Link assessment in an indoor 802.11 network," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2006.

[10] A. Wapf and M. R. Souryal, "Measuring indoor mobile wireless link quality," in *Proc. ICC*, Jun. 2009.

[11] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proc. ACM Special Interest Group on Data Communication (SIGCOMM)*, 2004, pp. 121–132.

[12] A. Tønnesen, T. Lopatic, H. Gredler, B. Petrovitsch, A. Kaplan, S.-O. Tücke *et al.*, "olsrd: an adhoc wireless mesh routing daemon," available online at http://www.olsr.org/.

[13] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. ACM MobiCom*, 2003, pp. 134–146.

[14] J. Janssen, D. De Vleeschauwer, M. Büchli, and G. H. Petit, "Assessing voice quality in packet-based telephony," *IEEE Internet Computing*, vol. 6, no. 3, pp. 48–56, May-June 2002.

[15] A. Lincoln, "Gettysburg address," audio narration by Michael Scherer, Americana Phonic, available online at http://www.loudlit.org/works/gettysburg.htm.