A Full-Scale Ad Hoc Wireless Network Test Bed

Timothy X Brown, Sheetalkumar Doshi, Daniel Henkel, Sushant Jadhav, Roshan-George Thekkekunnel
University of Colorado, Boulder, CO 80303
timxb@colorado.edu

There is a significant gap in careful studies of ad hoc wireless network behavior, especially in networks consisting of heterogeneous mixtures of fixed, low-mobility, and high-mobility nodes. This paper describes a full-scale ad hoc network test bed and its monitoring architecture. The test bed consists of fixed, vehicle mounted, and unmanned aerial vehicle mounted nodes. The test bed gives detailed data on network performance on a per packet basis under different operating regimes. These data enable us to better document and characterize ad hoc network behavior in complex real environments.

I. Introduction

In an ad hoc network, radio nodes either exchange packets directly or, if they are out of direct communication range, through one or more intermediate nodes that cooperatively relay the packets. Typically the nodes are mobile so that connectivity changes over time. This mobility is managed by specific ad hoc routing protocols designed to work well in dynamic network environments. Much research has investigated ad hoc networks.[7] But, the bulk of this research has been in theory and simulation with some limited attempts at real implementations.[6,8] These real implementations often differ from the simulations and theory.[2,3] There is a significant gap in careful studies of ad hoc network behavior, especially in networks consisting of heterogeneous mixtures of fixed, low-mobility, and high-mobility nodes. A comprehensive and efficient monitoring scheme can close the gap between real world and simulation. We need a scheme that comprehensively logs traffic information along with the node specific information so that every minute event that occurs in the ad hoc network can be accurately reproduced. The University of Colorado has developed a wireless network test bed using 802.11b radio equipment mounted on fixed sites, ground vehicles, and small low-cost unmanned aerial vehicles (UAVs). The test bed is located at a federal facility located near Boulder, Colorado. This paper describes the test bed, monitoring architecture and test results.

II. Test Bed

The test bed consists of the mesh network radio (MNR) nodes and the site where they are tested. The MNR is a compact package that consists of a Soekris single board computer, Orinoco 802.11b PCMCIA card, a Fidelity-Comtech bidirectional amplifier with up to 1W output, and a GPS. The nodes are packeged in environmental enclosures. These enclosures are mounted at fixed sites or on vehicles. The nodes are also mounted inside of a small UAV developed at the University of Colorado. The different mountings are shown in Figure 1. The MNR runs the dynamic source routing protocol (DSR)[4] communicating with other nodes via 802.11b. We chose DSR because its routing is on-demand. In on-demand routing, a traffic source only seeks a route to a destination when it has data to send. Thus, nodes do not waste bandwidth trying to establish routes they will never use. When a node needs to send a packet, it initiates a *route request* process among nodes in the network to establish a route. DSR also uses source routing whereby a packet source precisely specifies which route the packet will follow. We implemented DSR ourselves using the Click modular router.[1,5] With our own implementation we are free to modify the protocol as needed. The software runs under the Linux operating system (the WISP-Dist distribution*, a stripped down version of Linux whose size is 8MB) and has been ported to a number of other devices including laptop and handheld computers.

The UAV is a Telemaster-based design. The main criterion is that it be flexible for different flight configurations, have long (multi-hour) endurance for uninterrupted flight testing, and, have a large payload to carry the MNR and possibly other packages. The plane has a 15kg maximum takeoff weight and 5kg payload. Top speed of the plane is 190km per hour. The plane has been constructed here at the University of Colorado using carbon fiber composite construction techniques.

* leaf.sourceforge.net

(a) UAV (b) Fixed (c) Ground Mobile

Figure 1. Heterogeneous Ad Hoc Radio Nodes

The test bed is located at the Table Mountain National Radio Quiet Zone near Boulder, Colorado. The test bed is a 7 km^2 large flat area in which the use of radio transmitters is controlled. Table Mountain is operated by the Institute for Telecommunications Sciences. The experiments consisted of up to 5 ground nodes (mobile and fixed), 2 UAV nodes, and 2 laptop-based nodes. The test bed site is shown in Figure 2. It shows typical node placement, and roadways on and around the site used for testing.

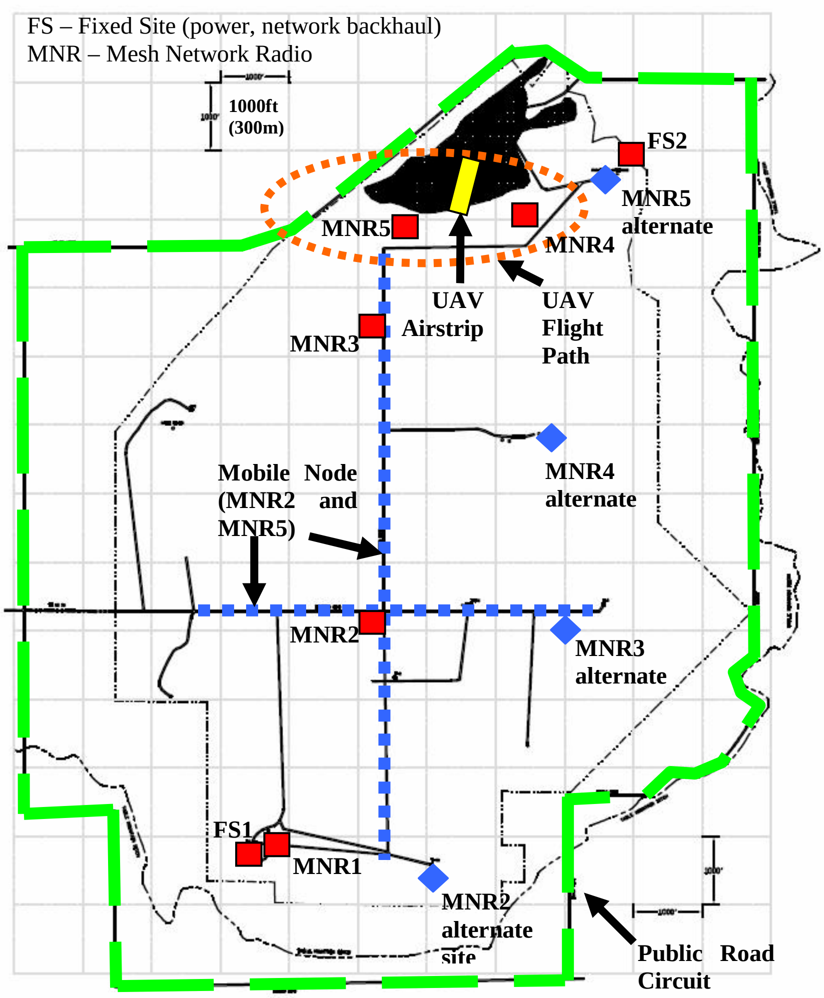


Figure 2. Table Mountain Test Site

III. Monitoring Approach

The monitoring must achieve several goals in order to be effective. The monitoring must provide sufficiently complete information to analyze network behavior in detail. The test bed data should be available in real time. The test bed should scale to 10's of monitored nodes. The monitoring should have minimal impact on the normal operation of the network. In reaching these goals, the monitoring must solve several challenges. The UAV radio nodes are small size, with limited power and payload. The ad hoc networking is complex with control distributed across the ad hoc nodes. Nodes may be disconnected for long periods of time during experimentation and the monitoring should be reliable to these disconnects.

These constraints limit some approaches. The real time collection requirement precludes simply storing monitoring data on each node to be collected after the experiment. The distributed behavior suggests that data has to be centrally collected and correlated between nodes. The scaling and interference constraints imply that the monitoring should use minimal computing, storage, and bandwidth resources.

The monitoring approach is shown in Figure 3. The pieces are described in the following sections.

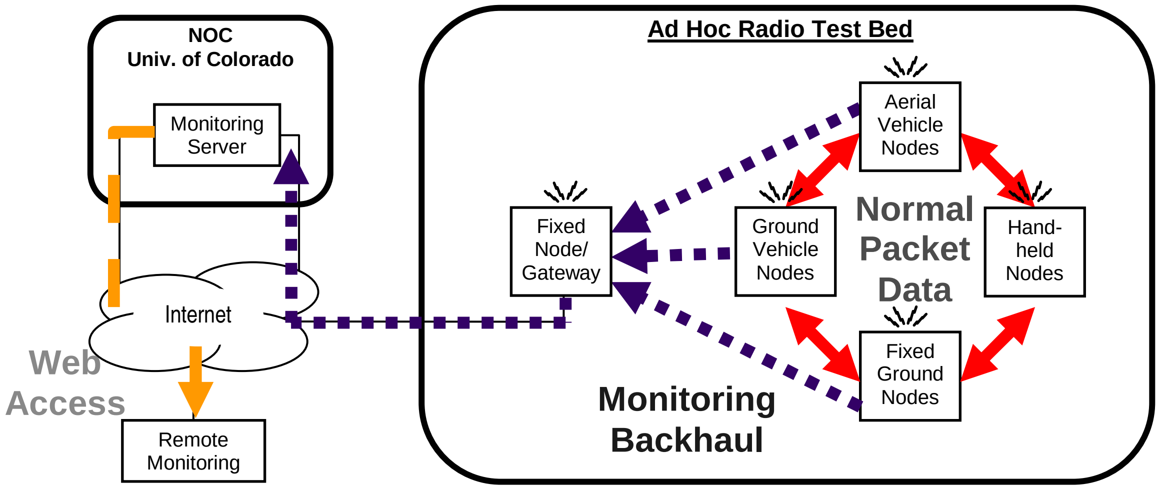


Figure 3. Normal data traffic (red solid) is monitored by each node. Periodically each node sends a report on the data (blue dotted) to the monitor server. This data can be viewed remotely over the Internet (yellow dashed) via a web-based GUI.

A. Radio Nodes and Gateway

We implemented DSR using the Click modular router.[1,5] This gives us great flexibility to monitor the network behavior. Running on each node is a monitoring process inserted into the radio packet processing as shown in Figure 4. The monitor collects data on every packet that is sent or received from the node. The information collected is routing level information which includes whether the packet is transmitted or received, the packet type (UDP/TCP/ICMP), packet route (available from the DSR header), time stamp, packet size, and packet sequence number (inserted by the packet source). Higher level application information from the body of the packet is not collected to minimize processing. Lower level 802.11 MAC and physical layer information is not available.

The monitoring also collects information from the GPS at one second intervals about UTC time, latitude, longitude and altitude of the node's current position. A unique feature of the monitoring is an interface for text messages to be time stamped and inserted to annotate the data during operation of the network. Through this mechanism, test scripts running on the node can send messages to the monitoring module so that packet-level behavior can be correlated with the higher level scripts.

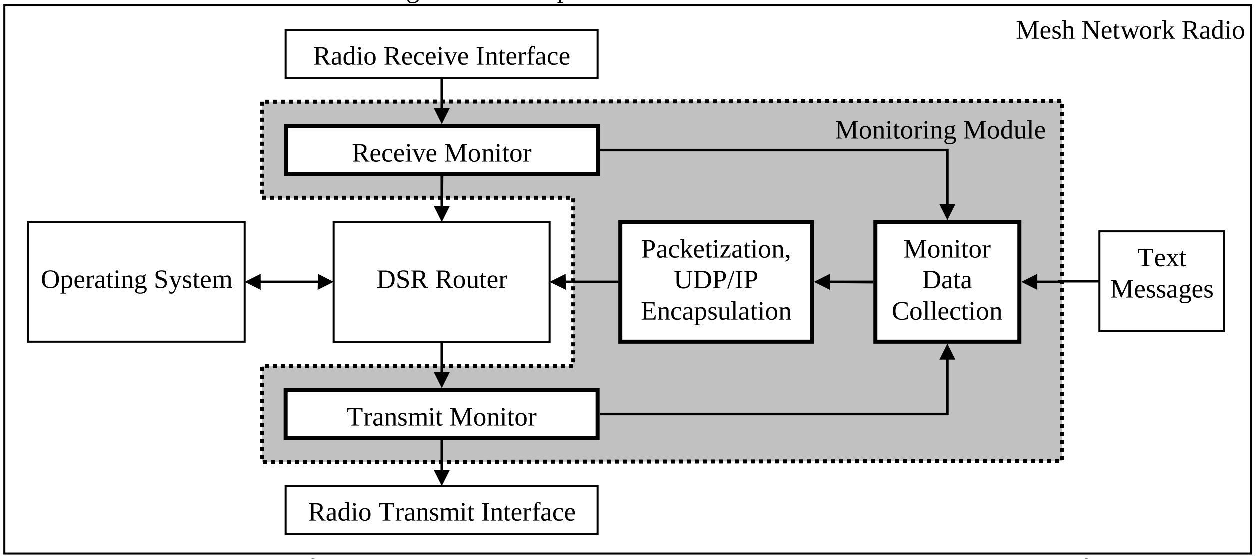


Figure 4. The monitor software (shaded) collects per packet data as packets pass in and out of the radio. This is periodically packetized and sent back to the monitor server.

The information collected by the monitoring module is packetized and a monitor sequence number is added to the packet which is unique per node. Packetization is triggered every 10 seconds or whenever the estimated packet size of the monitoring information equals 1000 bytes. This packet is sent to the test bed gateway.

The gateway may not always be available as nodes move and connectivity changes. Short (few second) outages can be handled reliably by standard network protocols such as TCP. But, a node on the test bed can experience outages of many minutes. For example some experiments may intentionally disconnect the network into one or more subgroups. The subgroups are communicating locally (i.e. there is something interesting to monitor) but may have no connectivity to the gateway.

To handle these outages, we developed a reliable monitor packet delivery mechanism. The monitoring module buffers each monitoring packet and a packet copy is passed on as an application layer packet to a module that adds to it a UDP/IP header with its destination as the gateway. This packet now is passed to the DSR router as a UDP/IP data packet and the DSR router routes this packet to the gateway node. The gateway node DSR router receives the DSR source routed monitoring packet, strips off the DSR header and recognizes the packet as a monitoring packet. It then sends back a Monitoring ACK packet to the node that sourced the monitoring packet. The node on receiving the ACK, removes the corresponding monitoring packet from its buffer and is clear to transmit the next monitoring packet it has lined up in the buffer. If the node does not receive the ACK packet, it keeps retransmitting the same monitoring packet until it eventually gets an ACK from the gateway for that packet. Each retry occurs every 10 seconds. If the packet is buffered for over 1 hour, the packet is dropped and the next packet in the buffer is passed on for transmission.

The gateway strips off the UDP/IP header of the received monitoring packet and adds a new UDP/IP header with the destination as the CU monitoring server. It then forwards on the packet over a wired interface which routes the packet to the CU monitoring server through the Internet.

B. Monitoring Server

The monitoring server receives the monitoring packets from the gateway, parses them, and inserts them into a database. The database is both a data archive and an analysis tool. The database stores three types of data, per-packet data, per-node data, and application messages. These three elements are each related to monitoring data to retain monitoring packet sequence numbers and timestamps on both packet creation and reception at the database.

The per-packet data is the packet data recorded at each node. Note that a single data packet will appear several times in the database since it is transmitted and received by different nodes on its path across the network. Each entry is associated with the point on the path where it was recorded. This level of detail enables a packet to be tracked as it crosses the network and either its successful delivery at the destination or the point where it was lost can be determined.

The per-node data is the GPS time and position data included in the monitor packet. The position of every node at every time during the experiment can be determined. In turn, the distance between any two nodes at any time can be determined. When combined with the per packet data, it allows packet losses to be correlated with node separations.

The application messages contain both free text and a numeric type to ease sorting and display. Examples include the start and stop times for experiments, the results of the experiments, and notification messages such as when a node powers up or the radio interface is turned off. By embedding this information in the database, the database becomes the complete archived repository of all test bed activities.

This information is stored in a central ODBC-compliant relational database. We are currently using MySQL[†] version 1.4.3 as our database engine since it is open source, freely available, and can be ported to many different platforms. The engine is hosted on a dual-processor 450 MHz Sun machine running SunOS release 5.8. The relational database enables complex queries for detailed network performance analysis. For instance in an experiment to see the effect of a UAV on ground node communications, all the packets can be classified into “not delivered”, “delivered via a UAV”, “delivered without a UAV”.

C. Remote Monitoring Graphical User Interface

The monitoring design also includes real time remote access and data visualization via a Web-based Graphical user interface (GUI). A screen shot of the interface is shown in Figure 5. The GUI is a Java applet (using version 1.4.2) using Sun’s standard GUI library Swing to display and analyze network state and performance both post-test and real time. The GUI shows the position of nodes and routes being used. Graphs versus time can be called up showing the traffic sent and received by a node; the traffic between two nodes, and the text messages in the

[†] www.mysql.com

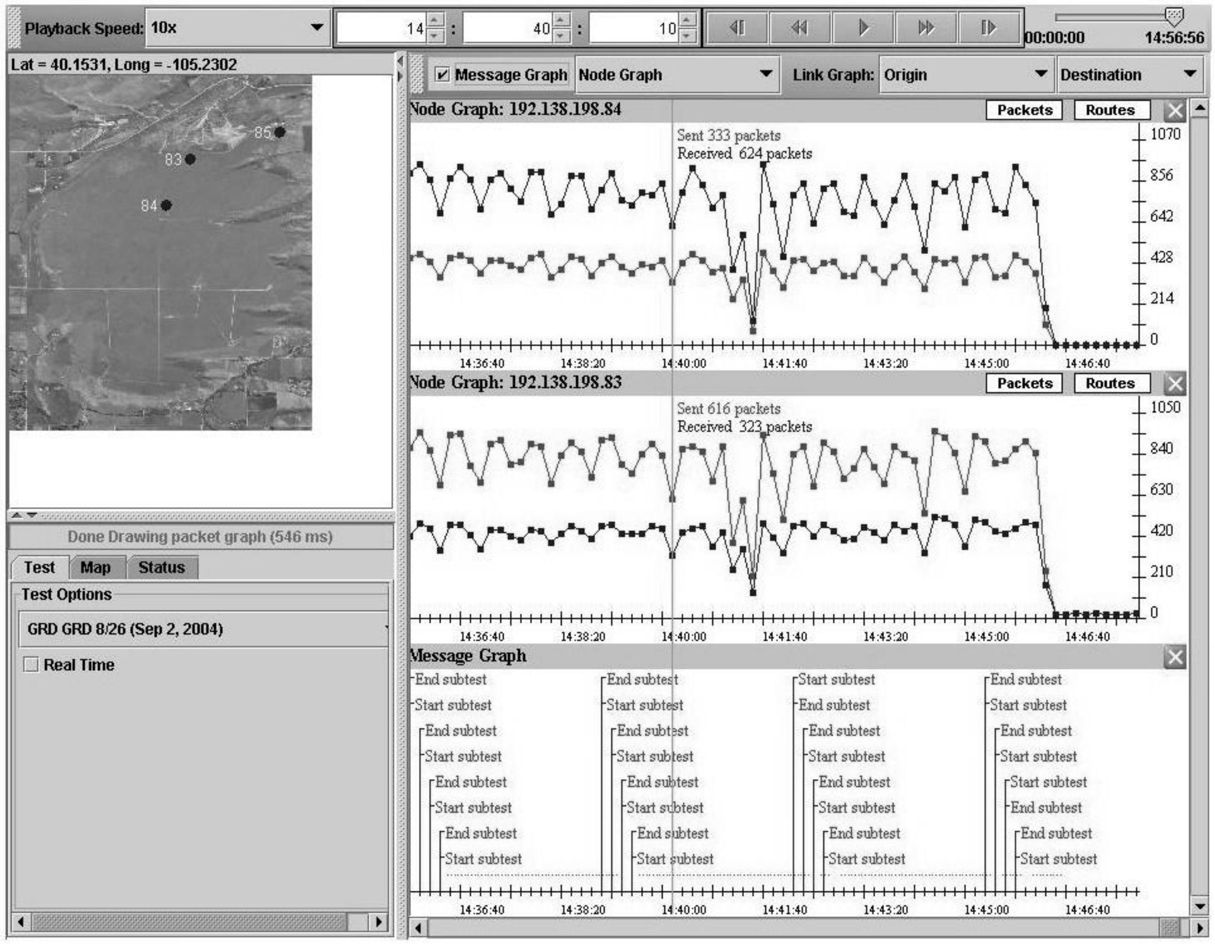


Figure 5. Screenshot from the remote monitoring GUI. Situation map is on the top left showing MNR locations at Table Mountain site. Status messages and control panel are on the bottom left. Performance and message graphs are shown on right. Time control is at the top.

database. All graphs share the same *x*-axis, and therefore the same time frame length and current position. This horizontal alignment of the graphs facilitates graph comparison. The traffic graph data can be filtered by the routes that packets take and by the packet types (TCP receive, TCP transmit, UDP receive, etc.).

The GUI serves several purposes; experimentation support, data dissemination, and data analysis. The experiments take place over a large area and situational awareness is limited. The GUI enables experimenters at the test bed site to observe node and traffic activity. For instance, when a radio and its GPS are properly functioning, they appear on a situational map in the GUI. Traffic and routing can be monitored during experiments for anomalies. By making the GUI Web-based, the data can be readily viewed by other observers and researchers. Finally, an ad hoc network has many simultaneous activities. The GUI provides a tool for comprehending the big picture and isolating specific events.

IV. Performance Results

The test bed performance was measured along several dimensions: The monitoring performance itself was measured along with initial data on the throughput, the effect of mobility, and subjective tests of performance.

A. Monitoring Performance

The first measure is the effect of the monitoring on network performance. For this test the utility netperf[‡] was used to measure TCP throughput of a one-hop (direct) and two-hop (one intermediate relay) path. These tests are

‡ www.netperf.org

demanding since they generate a high rate of data packets and acknowledgements that quickly fill up monitoring packets. The rate of monitoring packets in this case is about two per second. The throughput was measured with and without the monitoring and repeated 59 times. The average throughput along with the sample standard deviation is shown in Table 1. The data shows that monitoring reduces the throughput by 20-50kbps. This represents less than 10% of an impact on throughput.

Table 1. The effect of the monitoring software on throughput.

	Throughput (Mbps)	
	1 hop	2 hop
Without monitoring	1.40 ±0.02	0.53 ±0.12
With monitoring	1.38 ±0.03	0.48 ±0.17

The factor of 2.7 drop in performance going from one to two hops is typical of ad hoc networks and represents the double load placed on an intermediate relay node which must both receive and transmit every packet on the two hop path. At this point in our research, these are the kinds of effects that we are trying to observe and a 10% overhead for monitoring is acceptable. Future work is looking at compression techniques to reduce the size and frequency of the monitoring packets.

The monitoring module uses a reliable backhaul to deliver packets to the database. To see how effective this method is we plot the cumulative density function (CDF) of the delay from when a monitoring packet is generated to when it is delivered to the database in Figure 6. Half of the packets are delivered within a second, but, 5% of the packets required more than a minute and 0.1% required more than 10 minutes. Thus, the need for a dedicated reliable delivery mechanism is clear.

B. Throughput

The nodes are placed in a linear topology so that MNR2 to MNR3 is one hop; MNR1 to MNR 4 is three hops; and so on. These hop counts are nominal since the routing may discover shortcuts that skip nodes. But, generally these connections are weak with enough packet errors to force the network into the nominal number of hops. The throughput test measured the throughput between every pair of nodes and the throughput vs. the nominal number of hops computed. The test was repeated with a UAV mounted node. In this case sustainable shortcuts through the UAV can be formed since the UAV to ground range is longer. The point is to see if the network can discover these shortcuts and if they help throughput.

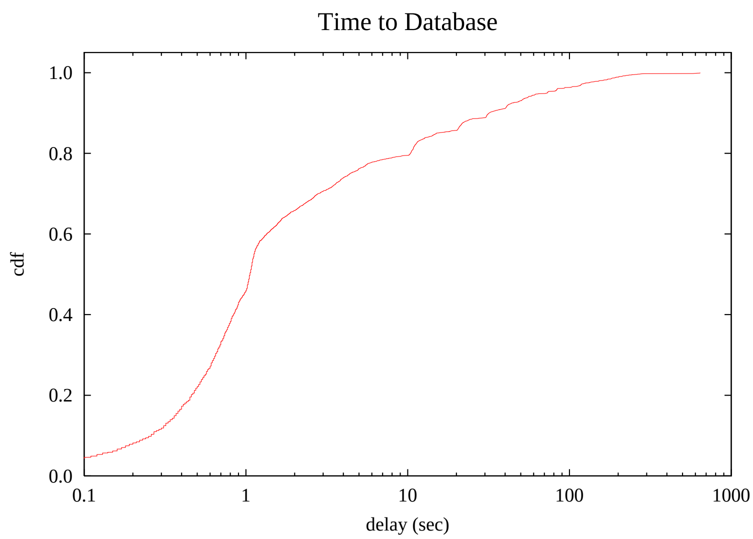
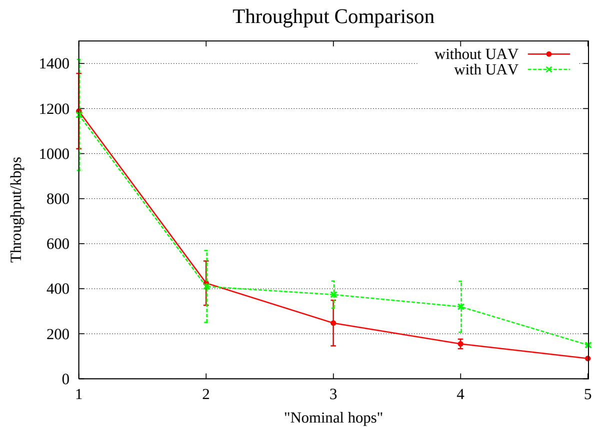


Figure 6. Probability monitor packet is delivered within a specified delay.

The throughput with and without the UAV is shown in Figure 7. The graph without the UAV shows the throughput falls off by a factor of two to three with each additional hop. This is a known phenomenon in ad hoc networks for a small number of hops. As expected the UAV makes no difference on one and two hop paths since paths between ground nodes that pass through the UAV are at least two hops. For three and four hop paths the UAV is able to maintain the throughput close to the two hop throughput indicating that the network is able to find and sustain the two hop paths through the UAV.

We note that the UAV throughputs have a higher variance on average. We attribute this to the UAV maneuvering. As the UAV turns it can bank away from a target node so that the ground node is not in the main lobe of the dipole antenna. To observe this effect we measured the losses in low rate packet streams sent over 20 second intervals between every pair of ground nodes with and without the UAV. The results are shown in Figure 8. Without the UAV, given the fixed stable and connected network, no packets are ever lost. With the UAV, shorter routes are formed through the UAV. Loss samples with the UAV experience occasional losses as high as 50% whereas the majority of samples have no losses at all.



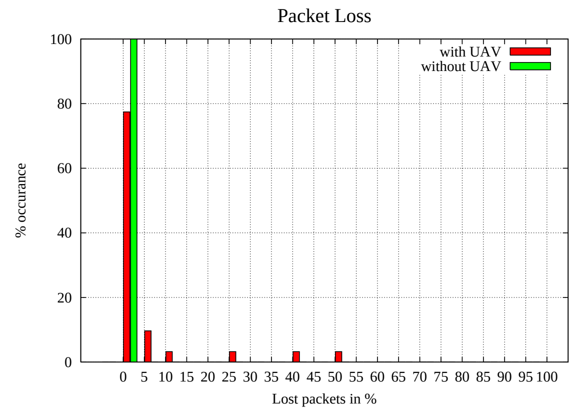


Figure 7. Throughput data with and without the UAV node vs. the nominal number of hops.

Figure 8. Loss rate over 20 second intervals with and without the UAV.

C. Mobility

The mobile tests divide the six node network into two groups of two adjacent fixed nodes separated by two mobile nodes. There are three classes of traffic in this scenario: within a fixed node group; traffic that has a mobile node as a source or destination; and traffic between the fixed nodes that is relayed by the mobile nodes.

Results for loss and throughput are shown in Table 2. The loss data shows that the UAV increases the loss rate in the mobile source destination cases, but decreases the loss rate in the mobile relay. The UAV flies on the north side of the mesa while the cars drive at the south side. Thus, links from mobile vehicles to the UAV are likely to be unstable and thus hurt the mobile source-destination scenario. Meanwhile, in the mobile relay case, the UAV can serve to avoid the mobile nodes altogether and thus it can provide a more reliable link for the fixed nodes. The throughput data shows that the UAV has a 10% negative effect on throughput in the first two cases, but, doubles the throughput in the mobile relay case.

Table 2. Loss percentage and throughput with and without the UAV in the mobile ground scenario

	with UAV		without UAV	
	% loss		%loss	
	mean	s.d.	mean	s.d.
within fixed node groups	6.2	12.2	0	0
mobile source-destination	45.8	36.1	10.8	25.9
mobile relay	17.5	17.5	35.8	35.3
	throughput (kbps)		throughput (kbps)	
	mean	s.d.	mean	s.d.
within fixed node groups	1120	363	1258	121
mobile source-destination	454	419	482	463
mobile relay	114	60	46	64

D. Subjective Tests

While measuring throughput and delay gives good quantitative values for network performance, it does not say much about the end user experience while using the network for day-to-day activities. Browsing the web, downloading files, or using streamed media is a very subjective experience which depends on every individual's habits and preconditioning. We designed two tests to capture subjective impressions: a web-browsing test and a voice quality test.

For the web-browsing test we have the candidate browse a website consisting of several pages with a different size image on each of them, namely 10kB, 100kB, 300kB and 500kB. The web pages are served by the smallfootprint, single-threaded web-server Boa (www.boa.org) installed on the gateway minimizing impact on gateway performance. Candidates report their experiences browsing the pages compared to browsing the Internet from their home connections.

The voice quality test evaluates the subjective perception of a voice conversation carried out between two test candidates using laptops associated to one of the nodes in the test bed or the gateway. The open-source, Linux-based SIP-softphone Linphone (www.linphone.org) proved to be stable and user-friendly. It supports several voice codecs and enables adjustment of SIP and RTP parameters to compensate for changes in network performance.

Preliminary experiments proved quite successful in terms of satisfying end user's expectations in a reliable, well-performing network. With a well setup network of fixed nodes browsing web pages from as far as six hops away can be compared to surfing the Internet on a fast dial-up connection. Rendering pictures gets more and more visible with increasing hop count, but is still acceptable. However, when changing positions within the network the delay incurred by finding a new route with DSR can be irritating. Also, sometimes downloads of pictures larger than 300kB stall noticeably halfway through the page, spoiling user's browsing experience. With a hybrid network of stationary and mobile nodes browsing becomes choppier as nodes move out of reach and new routes to the web-server have to be found more frequently.

Voice quality as tested from the gateway to a laptop moving around the test bed was found to be exceptionally good up to three hops and no noticeable end-to-end voice delay could be observed. New routes formed automatically and voice contact was re-established without having to re-dial or restart the phone application, although there were gaps in the speech. At a distance of four hops, voice streams became choppy and a meaningful conversation was not possible anymore. As seen with web-browsing, the time to discover a new route can also considerably impair voice conversations.

The impact of the UAV node still has to be investigated, but throughput and delay results suggest an improvement in user experience as well since most connections are three hops or less.

V. Conclusion

The test bed has proved useful in evaluating the ad hoc routing performance. It is able to show the value of UAV nodes while also clearly showing issues with the plane maneuvering. The monitoring architecture is able to reliably capture performance data even with long node outages while having only minor impact on performance. Testing and data evaluation is continuing and the final paper will include more detailed data.

The test bed is being used to evaluate the DSR protocol over the 802.11b MAC. The monitoring has no operational dependence on the MAC. The only dependence on DSR is the DSR packet parsing in the monitoring module and an optional sequence number header that was added to the DSR router. Future work will address other ad hoc routing and MAC protocols.

Acknowledgments

This work is supported through a grant from the L3-Comcept Corporation. Thanks to the Institute for Telecommunication Sciences for providing the Table Mountain site. Thanks to Gerald Jones and Jesse Himmelstein for support of the GUI interface. Thanks to Brian Argrow, Corey Dixon, Jack Elston, Jake Nelson, Philip Nies, and Bill Pisano for their part in the design, construction, and operation of the UAVs. Thanks to Daniel Henkel, Marc Kessler, and Roshan-George Thekkekunnel for testing support.

References

[1] Doshi, S. Bhandare, S., Brown, T. X , "An On-demand minimum energy routing protocol for a wireless ad hoc network," *Mobile Computing and Communications Review*, vol. 6, no. 2, July 2002.

[2] Feeney, L., Nilsson, M., "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment," *IEEE INFOCOM 2001*

[3] Heusse, M.; Rousseau, F.; Berger-Sabbatel, G.; Duda, A.; "Performance anomaly of 802.11b," *INFOCOM 2003*. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE, Volume: 2, 30 March - 3 April 2003 Page(s): 836-843

[4] Johnson, D., Maltz, D., "Dynamic Source Routing in Ad HocWireless Networks," *Mobile Computing*, Chapter 5, pp. 153-181, Kluwer Academic Publishers, 1996.

[5] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, August 2000. http://www.pdos.lcs.mit.edu/click/

[6] Maltz, D., Broch, J., Johnson, D., "Experiences Designing and Building a Multi-Hop Wireless Ad Hoc Network Testbed," *CMU School of Computer Science Technical Report CMU-CS-99-116*, March 1999.

[7] Royer, E., Toh, C., "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Communications*, April 1999, pp. 46–55.

[8] Sanghani, S. Brown, T. X, Bhandare, S., Doshi, S. ``EWANT: The Emulated Wireless Ad Hoc Network Testbed," *IEEE Wireless Communications and Networking Conference (WCNC)*, 16-20 March, 2003.