First-class meta-data: a step towards a highly reliable wireless seismic network in Peru*

Martin Lukac[1†], Igor Stubailo[2†], Richard Guy[1†], Paul Davis[2†], Victor Aguilar Puruhuaya[3], Robert Clayton[4], Deborah Estrin[1†]
Center For Embedded Networked Sensing[†], UCLA Computer Science[1],
UCLA Earth and Space Sciences[2], Universidad Nacional de San Agustin de Arequipa[3],
Caltech Division of Geological and Planetary Sciences[4]
{mlukac,rguy,destrin}@cs.ucla.edu,{stubailo,pdavis}@ess.ucla.edu,vaguilarpw@yahoo.es,clay@gps.caltech.edu

ABSTRACT

A 200-mile long temporary sensor network is inherently fragile, yet domain scientists legitimately demand reliable sensor data capture and transport. We report here on a currently deployed and operational sensor network system that considers sensor and general system health meta-data to be equally important as sensor data. We argue that this is both essential to, and effective for, reliable system operation. The meta-data constitutes a 0.1% storage space and transport system overhead in a fielded seismic sensing network.

1. INTRODUCTION

The Andean Seismic Project is a joint research effort by the UCLA Center for Embedded Networked Sensing (CENS) and the California Institute of Technology Tectonic Observatory (TO) to collect fine-grained long-period seismic data from four linear transects forming a rectangular slice of the southern Peruvian Andes. Local partners include Lima's national Geophysics Institute of Peru, Arequipa's National University of San Augustin, and Juliaca's Peru Union University. Very loosely speaking, the rectangle's corners are Cusco, Nazca, Arequipa, and Lake Titicaca; the transect here is the "southern line", which runs from the coast through Arequipa to the northern tip of Lake Titicaca. This is a region of special geophysics interest: the unusual Nazca plate and ridge lie just offshore in this subduction zone, the area is richly populated with active volcanoes, and to date, no high density broadband instrumentation has been devoted for an extended period in this area. This multi-year, multi-transect deployment effort is expected to provide a rich collection of data about the three-dimensional structure of the slab and its interaction with the continental plate that overlays the slab.

A parallel goal of this deployment is to test application-level ad hoc wireless network management techniques and tools in a difficult setting: medium scale (65 nodes), dispersed (nominal 6km intervals over 300km of alternating mountains and desert plains), medium term (18-24 months), with non-redundant network paths subject to many forms of disruption (component failure, weather, theft, livestock damage, preventive outages).

This paper focuses on the latter goal. It presents the architecture of our deployed system, valuable tools we have developed, and key lessons learned to date. We assess the current state, just a few months into full operation of the entire network following 18 months of planning and installation effort. Throughout, we compare and contrast this deployment with a preceding deployment of several years' duration in Mexico [13] [16] [17]. In particular, we have a goal of significantly increasing the percentage of usable data collected, in contrast to conventional seismic data collection methods, as a result of the integration of the seismic instrumentation into a wireless sensor network with Internet gateways.

Seismic research is frequently conducted by deploying a number of data logging seismometers. The challenge, especially with short- or medium-term deployments, is collecting high-quality data in the face of many logistical challenges. These challenges include: cost of regularly visiting each station to retrieve captured data; delays in noticing that data has been either incompletely collected or that the data was corrupted by sensor failure, misalignment, or other external source; and delays in rectifying problems, whether by adjustment or compo-

*The Peru array and this work were partially supported under NSF Cooperative Agreement #CCR-0120778 and by the Gordon and Betty Moore Foundation through the Caltech Tectonics Observatory contribution number 118.

ESSA Workshop '09, April 16, 2009, San Francisco, California, USA.

nent replacement. It is common for deployed stations to be visited on schedules that range from monthly to annually, depending on the funding available, with attendant latency in recognizing and repairing problems. Example problems include frequent sensor internal realignment to compensate for sensor base settling, GPS system failures that interfere with precise timestamps, flash card failure, and power system failures. We claim our wireless networked approach dramatically reduces these delays from months to hours and in many important cases, it can prevent data loss by enabling recognition of system health trends that indicate impending failure and allowing targeted preventive maintenance.

In contrast to many embedded sensor networks, our network deals with continuously sampled data (100Hz) with an "every bit is critical" integrity constraint. We also have a power concern: each station consumes up to 5W continually, of which half is sensor/digitizer, and half is for data storage and transmission.

The current deployment builds on a prior one that spanned the Mexican isthmus at Mexico City and operated from 2005 to 2007. The physical components are largely the same in both deployments, but the software has undergone significant enhancements driven by operational issues experienced in Mexico. In particular, we have dramatically increased logging of a wide range of system health parameters; we have developed visualization tools that allow rapid analysis of logging data, and encourage regular use by stakeholders; we created a suite of tools to assess data quality; and we wrote better ad hoc network topology analysis tools to improve dynamic data transport routing decisions.

These improvements had an immediate impact on how we manage network operation and maintenance, even prior to its completion. The result is significant improvements in data quality and completeness at an unprecedented early stage. We also have a rich meta-data set to be used in the future to assess the impact of system faults on sensor data quality, identify common failure modes, and propose changes in operational methods and component selection. To our knowledge, this system's wealth of meta-data and ability to interactively control seismic instrumentation is unique.

We believe our experiences have broader applicability to sensor network deployments in general, and especially to very large deployments such as the planned NEON sensor networks across the USA [3].

2. TRANSECT ARCHITECTURE

In this deployment, the transect is a collection of forty-nine seismic sensing stations placed at roughly 6km intervals along a line running from the ocean port of Mollendo, across eight plains and seven mountain ranges, and ending 20km northeast of Juliaca near the shore

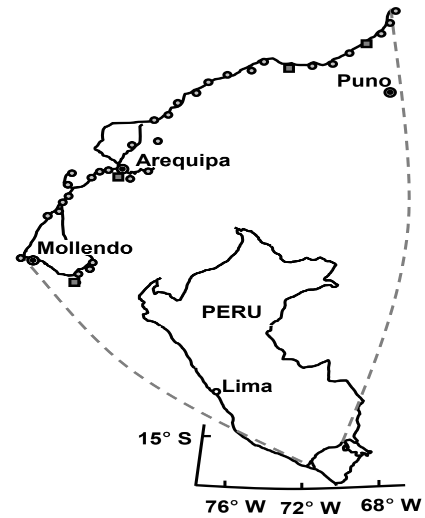

Figure 1: A map of the Peru deployment.

of Lake Titicaca. As with our previous deployment in Mexico, we anticipated that topography would naturally divide the line into various disjoint wirelessly-connected segments, each of which would be "anchored" by a conventional Internet gateway node, such as a PC with a DSL connection. While we hoped for all of our stations to be nominally accessible from UCLA, we expected that a small number would not be connected; 47 of 49 deployed sensor stations are connected.

Each station has a CENS designed and fabricated "CENS Data Communications Controller" (CDCC) with two basic functions: it executes the data collection software that interacts with the digitizer and sensor and it executes the software that transports data onwards in the wireless network towards the gateway sink node (an internet connected PC RAID). A CDCC contains a low-power microserver with an ethernet interface, compact flash card for data storage, and a PCMCIA radio card with external antenna port connected by cable to a pair of directional antennas and optional amplifier. In Peru, our segments range up to 28 stations plus additional repeaters, with a diameter of up to eight wireless hops.

In this type of deployment, "broken" is normal: there are enough components, and sufficient "hostile reality" aspects to defeat conventional end-to-end protocols such as TCP. To be successful, we need protocols for both data transport and also system health monitoring and control that do not fundamentally rely on realtime, end-to-end connections. A major contribution of this work has been the development, deployment, and effective demonstration of our Disruption Tolerant Shell (DTS) and related protocols [16]. With these tools, we can routinely operate the system when network links are very poor quality, or even failed much of the time.

An often-underused facet of wireless sensor network

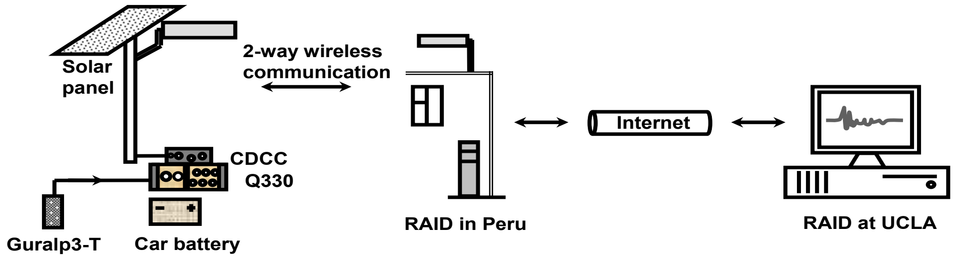

Figure 2: A diagram of our typical setup.

operation–but which is critical to our deployment–is the use of pervasive system health logging. We collect extensive meta-data from the sensor, the digitizer, and the CDCC (yet only a tiny fraction of the seismic data volume) and process this metadata at UCLA. With suitable display tools, this data can be used in near real time to diagnose current and impending problems, and in crucial cases *resolve* the problems. Effective logging and monitoring is key to our goal of significantly increasing the percentage of usable seismic data collected.

3. SITE SELECTION

Deployment begins with site selection, once a nominal transect line has been established. Major considerations include security, accessibility, and RF visibility to adjacent sites; the latter is a significant constraint in a topography rich locale, and makes site difficult when compared to conventional seismic deployments.

In most cases, the presumption is that a site will need to communicate both downstream (to send its data towards an internet gateway) and upstream (to receive data from a neighbor). Our Mexico experience taught us that a 200mW 2.4 GHz radio, 3m mast, and 15dB YAGI antenna are sufficient for most line-of-sight paths up to 15-20km over level ground, and up to 30km hilltop to hilltop. When coupled with a 1W amplifier and 1m antenna cable, up to 50km is feasible.

Our Peru transect serendipitously includes a 20,000ft volcano peak about which hosts a forest of antenna masts at 16,000ft that provide a 270-degree view. We exploited these existing mast sites by arranging with the owners (private firms and government agencies, some needing year-long negotiation) to use their masts for our antennas, often at 100' elevation. These prominent antenna placements allowed us to construct local star topologies with dedicated long-haul links (30-50km) between several hubs. The resulting overall topology is 5 disjoint internet-linked wireless segments: three short chains (of 3-11 sites each), one singleton, and one grand segment of 28 nodes organized as three mountaintop hubs linked to a single internet gateway in a valley below. The density of RF activity at these hubs prompted use of all three non-overlapping WiFi channels to reduce congestion.

3.1 A typical site

Every modern seismic station is comprised of a sensor, a digitizer, and a data logger. In our deployment, seismic motion is sensed and converted into a digital data stream by a broadband Guralp 3T seismometer connected to a Kinemetrics Q330 digitizer with a short cable. The digitized output stream then is sent to a (CDCC) that incorporates a low-power processor module (Intel/XBow Stargate), a 4GB compact flash storage card, and a 200mW WiFi radio. The seismic signals are continuously sampled at 100 Hz with 24 bit resolution per channel, yielding 25-50MB/day of compressed data. The 4GB local storage is sufficient to tolerate communications interruptions of 80-150 days, although in most cases the data reaches an internet-connected PC RAID within two hours of capture. 60% of the sites are solar powered; 40% receive power from 220V mains. All sites incorporate either a 12V 100Ahr gelcel or marine deep cycle battery, to provide a multi-day reserve for the 5W load presented by the site, when (not if) power source interruptions occur.

We learned from our Mexico work that radio reception can vary dramatically in the space of a few meters, and so it is important that the site components be placed exactly where the site planners specify. This allows for the use of shorter antenna cables leading to markedly less signal loss and ultimately to more robust wireless links. For example, a 10' cable has one-fourth the signal attenuation of a 100' cable, so there is great incentive to reduce cable length.

3.2 Internet connectivity

Our interest in internet connectivity is two-fold. The obvious interest is the ability to receive seismic data at UCLA in real time (2-4 hour delay) from Peru. Equally important is the ability to monitor and control the network in Peru *from UCLA* as a means of ensuring real time sensor data delivery. This is harder that it seems, in part because only one (of five) of our internet connections includes a static IP address (alas, not the one serving 60% of our sites). To support interactive monitoring and control of the network, we found it necessary to provision the internet connection nodes (three PCs, two CDCCs) with lightweight application-level virtual private networking (VPN) software [2] that presents a private static IP assignment space suitable for supporting ssh and other standard Linux tools.

4. SOFTWARE SYSTEM ENHANCEMENTS

The software system in our Peru deployment builds on the system used for three years in the 2005-2007 MASE deployment in Mexico [13]. Key parts of this system, beyond the Linux environment, include a special command shell, a replicated state synchronization service built on the CENS Emstar sensor network deployment system [11], a link quality estimator, a loop-free sink tree routing protocol, and a custom hop-by-hop file transfer mechanism. The Disruption Tolerant Shell (DTS) [16] provides a powerful *asynchronous* command shell to enable a user to control network nodes with the issuance of a single command. It enables the user to check status, issue commands, and distribute files on all the nodes in the network or any individual node. Currently unreachable nodes (due to link quality or hardware faults) will eventually receive the commands and report the results to the entire network. DTS is layered on Emstar's StateSync [10] replicated state synchronization service. StateSync provides a high level publish-subscribe interface to a low-level tool that manages replicated state in an optimistic, softstate fashion.

A critical aspect of ad-hoc network operation and management in hostile environments is assessment of link quality–especially when alternate paths might be available–and selection of the "best path" between source and destination. Our experience with the MASE deployment, lab tests, and the experiences from EAR [14], led us to switch from link estimation based on ETX [7] to an algorithm based on ETT [5] that we have deployed in Peru. The system also uses a simple sink tree routing protocol with full paths to the sinks published up the tree to prevent routing loops. This routing protocol also leverages StateSync to disseminate the full path information to neighbors.

The data acquired from the Q330 digitizer is stored in one hour long files. Each file is bundled with state and handed off to the rest of the system to be delivered to UCLA. Data bundle transfer must be done hop-by-hop in our partially-connected network environment. Due to the number of hops and the fragility of network links, some nodes will never be connected in real time to their respective storage servers (even the ones in Peru), so we use a hop-by-hop file transfer mechanism that supports resumption of partial file transfer following recovery from a network link failure.

Our three-year experience with the MASE deployment convinced us that our basic approach was effective, but insufficient. In particular, our approach to network management was primarily driven by the success (or failure) of receipt of hourly data files at the UCLA-based storage server from the 50 networked sensor stations: if the UCLA server is receiving data relatively quickly (say, within 24 hours), then all is well. Conversely, if data does not arrive in a timely fashion, then something is *broken* and likely requires human attention in the form of realigning antennas, replacing failed hardware, or tracking down and fixing a software bug.

This approach worked to some extent. At the outset, we did not see a need for regular system-generated reports on system status, or the need to archive and study such reports. Such information, when we did attempt to obtain it, was collected using DTS to query various nodes about their current status, occasionally remotely connect to an individual station, or actually visiting the stations in person and using a laptop to connect (by wire or radio). Our methods did seem effective since we can infer a number of issues which affected the data flow, however there were two main problems with this approach. The first was that this method lacked timely visibility into data quality issues: as we discovered after the deployment, the system time could be off due to GPS system problems, the channels in the Guralp seismometer could be locked, or the masses in the sensor may need centering. These issues make the data unusable–a situation that one must discover and remedy immediately, since there is little point in collecting garbage! In Mexico, we were aware of the timing related data quality issues, but we did not fully understand the source and the scope until the end of the deployment, at which point it was too late to compensate in any way. Timing issues are discussed in [15].

The second problem was the cost of human time expended to figure out why data was not being received and the time spent to attempt to repair the issues. Some of our software tools were designed for computer engineers, not seismologists and geotechnical personnel, and proved to be difficult to use without post-graduate training in computer networking and operating systems. Some of our nodes were a three-day round trip from the operations center at the middle of the transect in Mexico City, so it was difficult to know what spares (and how many) to take along, which tools, etc. These time costs contribute to gaps in the data stream, which are also problematic to domain scientists: some studies depend on long-term continuous data streams, but others depend on the total duration of data collected, and so outages lead to increased deployment duration and costs.

A key goal for the current Peru project is to overcome these issues by enabling a better understanding of what is happening in the network, and in turn, to improve the overall data quality and completeness. We want to provide a near real-time, higher granularity, *end to end* picture of what is (not) working and to what degree. This includes tools to help improve the usability of the software system, so that domain scientists can better manage the overall system.

This section highlights four main areas of our system we focused on for improvement: logging, visualization, data quality control, and system usability. The system now does extensive logging of a variety of sensor, digitizer, and network parameters. It now includes database visualization tools that use both tabular and graphical representations of log data, and contains log analysis tools that alert operators to potential problems in the data quality. The system tools are usable (and used) by domain scientists lacking computer science training.

4.1 Logging

Logging is a standard requirement in embedded sensing systems. When we began the MASE deployment, it was unclear what the most useful information to record would be. As discussed above, we originally felt we could figure out most problems just based on whether we were receiving data or not; however, this was not the case. For the Peru deployment we wanted to log as much as possible without overloading the system and significantly changing the software stack. We built a simple logging component: it accepts messages over an IPC channel from any other component in the software stack. Each component issues log messages along with its identification and the logging component timestamps and buffers the message. The logging component buffers new messages for a minute, then writes the message along with any messages that have come in to disk. Every hour a new log file is started and the previous file is compressed and placed alongside the sensor data to be transferred back to UCLA where they are immediately parsed and inserted into a database. The compressed files average less than 1.5KB per hour (cf. data files of 1.5MB/hour).

Most of the components in the software stack output a status message every 10 minutes. For example the most recent status information from the Q330, the current link quality information, the most used data rate, the current SNR, and the total and used disk space. Other processes in the software stack only write log message on events. For example, the file moving component, which decides which files to send to a neighbor and monitors when incoming files have been successfully received, outputs a log message upon a successful receipt. The disk management component, which monitors the disk space and deletes old files to make room for new ones when the disk space has reached a certain limit, outputs a log message every time it deletes a file. Reboots are reported in the log by writing a log message every time the software system starts.

Site	2009-03-27	2009-03-28	2009-03-29	2009-03-30	2009-03-31	Curr CDCC
PE01	0 / G	0 / G	0 / G	0 / G	0 / G	
PE02	0 / G	0 / G	0 / G	0 / G	0 / G	
PE03	24	23	24	24	3	185
PE04	24	24	24	24	2	161
PE05	24	24	24 / 1	24	2 / G	41
PE06	24	24	24	24	2	153
PE07	24	24	24	24	1	141
PE08	24	24	26 / 1	24	1	210
PE09	24	24	24	24	2	98
PE10	24	24	24	24	1 / G	93
PE11	24	24	24	24	2	63

Figure 3: The main grid showing the number of files per day for the first 11 stations on March 31st, 2009. A red number indicates there is a reboot. An orange G indicates there are no GPS locks. A blue number (not shown) indicates a sensor command. A green number (not shown) indicates a file system error. PE01 and PE02 are stand alone stations so no data is shown until data is manual retrieved from the stations.

Logging is most useful when combined with visualization, which is provided by our web based interface to the log database. Logging and visualization create the ability for us to detect a number of common faults: no gps locks, un-centered sensors, poor SNR, power problems, and a routing bug. These problems affect data quality or the ability to collect and deliver data.

4.2 Visualization

Throughout the MASE deployment we focused on whether data was being delivered successfully. We used a script on the archive server to determine which data was successfully delivered. Due to poorly understood varying delays in data delivery, the script was used irregularly and with a focus on delivery gaps several days old. Even though we had one or two full-time personnel in the field, often days would pass before realization that a real failure existed. This approach contributed to the length of gaps in the data stream.

We now provide the same information and much more through a web application which provides a front end to all the log data in the database. The most used page is what we call the main grid, in which each row is a station and each column is a day; the start date and the number of days to display is user-selectable. Each grid square shows the number of files received for that day and adds information to the square if there are no GPS locks for that day, if there were any reboots, the number of sensor commands (unlocks or centers), and number of file system errors. A screenshot of the main grid is shown in Figure 3. As with our previous deployment, the number of files received per day is expected to be 24. If there are less or more (e.g., perhaps caused by frequent reboots), we have an idea if the network or power system is having problems. Knowing if there were no GPS locks tells us whether the GPS system is connected and working and assures use that the data is properly time stamped. [1] The number of reboots lets us know as soon as any station begins to have power problems. Early on this information, combined with the number of files per day, let us quickly discover that several batteries were failing sooner than expected.

As another prominent (and painful) example, in preparation for this deployment we purchased new CF cards for each node. Shortly after installation, on a number of the cards the filesystem became corrupt. Reformatting would temporarily repair the problem, but the cards eventually degraded to the point where even the partition tables were unreadable. These cards have been replaced with lower density ones from a different manufacturer. To spot other CF cards developing this problem, the component which decides which files to transfer to the next hop reports if there are any I/O errors or another types of errors reported while obtaining the file size and reading the first few bytes of the file. The component attempts to delete the files if it can. Since the degradation of the CF cards seems gradual, using the logging system to report the problems provides an early warning of the larger problems to come.

From the main grid, selecting a particular day for a site displays the inspection page. It shows the creation timestamps and sizes of the data files received, the average disk space statistics, the average link quality and SNR to each neighbor, the unique paths to the sink, the time of each reboot if any, the time of each sensor command if any, and any files that were deleted. The inspection page also provides some options to display information from the logs. We can trace the path of each data file and look at a plot of the link quality and SNR to each of the neighbors for that entire day.

Also from the main grid, by selecting a particular site we can look at plots of the last three months of the sensor mass positions, the input power in volts, the Q330 internal temperature, current used by the Q330, and the GPS antenna current. In addition to looking at these values for each station individually on a single page, we have a page available which shows sparkline versions of the mass positions, power, and temperature for each station for the last two months all on one page. A screenshot of the sparklines for two stations is shown in Figure 4. The sparkline plot provides an incredibly quick means to identify common problems. We have been able to discover sensors that need to be centered and we have found two sensors each with a single channel permanently locked which need to be sent off for repair. We have also been able to spot worn out batteries or charge controllers that are failing.

[1] A major issue in MASE was belated discovery of antenna cable problems traced to weedcutters and corroded connectors. We can now see/identify these problems in real time, and have had one opportunity so far in Peru to leverage this information–an errant hoe slashed a GPS cable.

4.2.1 *Driving debugging with log information*

Early in the Peru deployment, we successfully used log information to discover a routing bug which was resulting in the loss of a data file every few days. The information showed that the missing data was being sent backwards up the sink tree one hop before disappearing from mention in the logs. This prompted us to ask two questions which directed our search for the problem: why were the files being sent up the tree, and why was the file being deleted when it was sent up the tree. We found three bugs: a typo in the length of a timer (18000 milliseonds vs 180000 milliseconds), an routing bug where a next hop was chosen with out considering whether the next hop had a path to the sink, and a missed internal state transition. The first two would cause a file to be erroneously sent back up the tree after a reboot, and the third would cause that file to be deleted. Without the logs, investigating the disappearance of data files would be much harder.

4.3 Data Quality Control

There are several levels of data quality control. The first is determining whether there is data. In the MASE deployment we determined how much data we were receiving through a script which ran on the main server and listed the number files per site per day. For Peru, as the data is processed on the main archive server, a log is generated which is inserted into the same database which holds the station log data, and a table is used in the web interface to show the number of files per site per day. The information obtained from the web interface is equivalent to the information obtained using the MASE scripts, just more convenient.

The next level of data quality control is determining whether the time synchronization of the data is good. We know that the time synchronization is good if the Q330 digitizer was able to obtain a GPS signal. If it can not obtain a GPS signal, the clock may be drifting and power problems may cause large time offsets. If there are no GPS locks, we know from MASE experience that the antenna is unplugged, the cable is broken, or the Q330 is misconfigured. The regular Q330 status reports include the GPS data and are logged every 10 minutes. The main grid display will show if there were no GPS locks for a station on a given day.

The third level of data quality control addresses potential problems with the Guralp sensor. The Guralp 3T is an active accelerometer that report the position of the mass for each of the three channels. The Guralp can lock masses in place for transport and can also center the masses to compensate for slight changes in the level of the seismometer which cause the masses to drift over time. During the first few months of the deployment while the cement in the sensor vault base settles, a

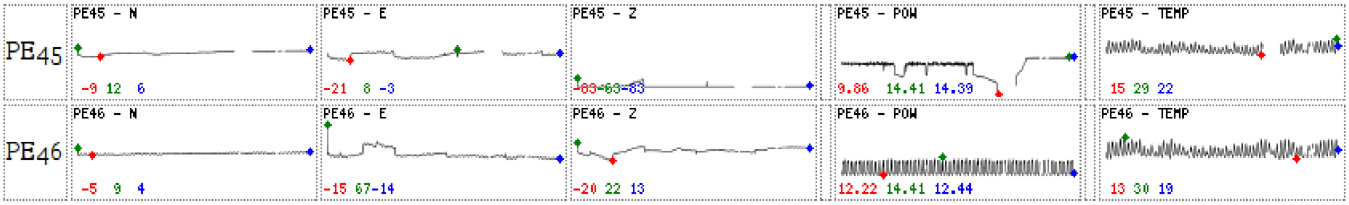

Figure 4: Sparklines showing the mass position, power, and temperature for PE45 and PE46 for February and March of 2009. The red, green, and blue numbers and marks indicate the minimum, maximum, and most recent values. The mass on the Z channel on PE45 is at its limit and requires centering. PE45 suffered from a power problem during the beginning of March resulting in log data gaps.

mass may repeatedly reach the limits of its movement. Throughout the rest of the deployment, temperature and other factors may contribute to poor mass placement as well. In these situations the sensor output on the channels is invalid since the mass movement is restricted. The status information written to the logs by the acquisition software contain the mass position for the three channels. The web interface can generate a plot of the mass positions for any sensor over any period of time. As described earlier, the web application provides sparkline plots of the last two months for the mass positions, power, and temperature are generated daily and all displayed on one page (see Figure 4). After first use of the web interface to view the mass positions on a number of sensors shortly after installation, we realized the extent to which we had mass position problems and so we added a script to each station to automatically issue a center command through the Q330 once a month.

The final level of data quality control is analyzing the actual waveforms from each sensor channel. This is routinely done after the data files are delivered at processed at UCLA, but in principle can be done in situ with a laptop and suitable software–and sufficient experience in reading and understanding seismograms to enable identification of problems such as locked masses or incorrect positions. Part of our future work is implementing automatic data quality analysis by searching for regular known signals such as microseism.

4.4 System and Network Usability

On major issue with the MASE deployment was that the interface to the network and the system was entirely through the CDCC's Linux command line interface. This meant training everyone involved with deployment and maintenance, and even with training, those with years of experience working with Linux (but little or no field experience) had a much easier time using the system and diagnosing problems. Both field technicians and those at UCLA expressed frustration. For Peru, we wanted to address the in system usability for both the field operators and the remote operators.

In response, we set up a lightweight embedded webserver to run on each node, reasoning that a suitably designed web interface would be much more usable. The web server provides status information as well as configuration options that are available through the command line interface. The status information provided is link quality information, disk space information, whether the time on the node is correct, sink and path information, a list of transfers in the past 5 minutes, and the status information from the Q330. The webserver also provided the ability to alter critical configuration parameters (for example, the Q330 serial number is used as a protocol address), instead of using the command line interface to hand edit a configuration file.

In addition to addressing the system usability, we wanted to improve the network usability: in particular, to simplify the processes of determining whether wireless links are good and stable. We learned from the MASE deployment that a nominal linear physical topology rarely resulted in a linear network topology, due to widely varying terrain. In practice, such deployments are inherently three-dimensional, and the resulting workable wireless topology was commonly nonlinear. This reality meant that establishing the wireless network required extensive testing of alternatives by pointing the (directional) antenna in various directions and verifying that a link could be established. However, this was typically done by simply "ping-ing" the remote site, noting the delay, and concluding that a low RTT indicated "success". Occasionally, the deployer would use a test transfer to verify to a higher degree that the link was satisfactory. Unfortunately, even the latter step of a test transfer is insufficient, as it does not account for congestion resulting from concurrent transfers.

To address network usability issues experienced in MASE, we rewrote the link quality estimator to use ETT instead of ETX. ETT provides a better stable reading of the link quality because it uses unicasts and the transmission rate. Our lab tests verified all the improved qualities of ETT. The switch to ETT means that we could simply look at the success rate and the transmission rate to determine whether the link was good enough. The embedded web server displays this information along with the SNR.

Overall, the embedded web server and the new link quality estimator have not been successes. The embedded web server was useful mostly for setting the Q330 serial number during the initial deployment push when there were two teams deploying the nodes. Since then it has fallen out of use. The link quality estimator does provide accurate estimates of the link quality but has rarely been used in the field to verify a link is good. The immediate feed back provided by test transfers has been used almost exclusively. Future work will involve creating an interface through the embedded webserver to do test transfers. It must provides real time feed back on the data rates achieved on a link.

5. RELATED WORK

The International Monitoring System [9] is a sixteen element broadband seismic array used to examine the North African Craton Structure and Seismicity. All sensors are placed in 50 m boreholes in the circular array of 6 km diameter (three rings with a central element) and are connected wirelessly in a star configuration. SOSEWIN [20] is a wireless network of units comprised of low-cost components with hierarchical alarming and routing protocols used for locally pre-processed data delivery. Two separate networks have been deployed to monitor volcanos [19] [18]. These are smaller scale systems with different requirements, different hardware platforms and different radios, but the software systems all attempt to achieve the same goals. There are a number of long distance wireless networks for research, remote regions, and developing nations [6] [1] [8]. These are focused on bringing Internet connectivity to remote regions and enabling delay tolerant networking research. A portable wireless network for monitoring weather conditions in wildland fire environments [12] used specialized point to point and point to multi-point radios designed for long distance wireless as the back haul link connecting a number of clusters of sensor nodes made up of motes and web cameras. A summary of insightful deployment advice and experiences is complied in [4]. Their collection of deployments differs primarily in geographical size and diversity, deployment length, and platforms used.

6. CONCLUSION

We have presented the arguments and architecture for a 50+ node reliable sensor network system that spans 200 miles of desert plains and mountains in southern Peru. Key to successful high reliability operation is the collection and analysis of pervasive meta-data about the sensor itself and many other general system and network parameters. We believe that this approach will serve as a useful model for future large scale and geographically dispersed sensor networks. Please visit http://peru.cens.ucla.edu for deployment pictures!

7. REFERENCES

[1] Hpwren. http://hpwren.ucsd.edu/.

[2] n2n vpn software. http://www.ntop.org/n2n/.

[3] National ecological observatory network (neon). http://www.neoninc.org.

[4] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli. The hitchhiker's guide to successful wireless sensor network deployments. In *SenSys '08*. ACM, 2008.

[5] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *MobiCom*, pages 31–42. ACM, August 2005.

[6] K. Chebrolu, B. Raman, and S. Sen. Long-distance 802.11b links: performance measurements and experience. In *MobiCom*, 2006.

[7] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Mobicom*. ACM, 2003.

[8] M. Demmer and K. Fall. Dtlsr: delay tolerant routing for developing regions. In *NSDR*, 2007.

[9] C. H. Estabrook, B. H. Bergsson, S. Soumana, O. Boureima, and M. Moumouni. Results from ims seismic array in niger. In *Eos Trans. AGU Fall Meet. Suppl*, number S43D-1905, 2008.

[10] L. Girod, M. Lukac, A. Parker, T. Stathopoulos, J. Tseng, H. Wang, D. Estrin, R. Guy, and E. Kohler. A reliable multicast mechanism for sensor network applications. Technical Report 48, CENS, April 2005.

[11] L. Girod, N. Ramanathan, J. Elson, T. Stathopoulos, M. Lukac, and D. Estrin. Emstar: a software environment for developing and deploying heterogeneous sensor-actuator networks. *ACM Transactions on Sensor Networks*, August 2007.

[12] C. Hartung, R. Han, C. Seielstad, and S. Holbrook. Firewxnet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments. In *MobiSys '06*. ACM, 2006.

[13] A. Husker, I. Stubailo, M. Lukac, V. Naik, R. Guy, P. Davis, and D. Estrin. Wilson: The wirelessly linked seismological network and its application in the middle american subduction experiment. *Seismological Research Letters*, 79(3), May/June 2008.

[14] K.-H. Kim and K. G. Shin. On accurate measurement of link quality in multi-hop wireless mesh networks. In *Mobicom*, 2006.

[15] M. Lukac, P. Davis, R. Clayton, and D. Estrin. Recovering temporal integrity with data driven time synchronization. In *IPSN 2009*, 2009.

[16] M. Lukac, L. Girod, and D. Estrin. Disruption tolerant shell. In *ACM SIGCOMM CHANTS*, Pisa, IT, 2006.

[17] M. Lukac, V. Naik, I. Stubailo, A. Husker, and D. Estrin. In vivo characterization of a wide area 802.11b wireless seismic array. Technical Report 100, UCLA - CENS, 2007.

[18] W. Z. Song, B. Shirazi, S. Kedar, S. Chien, F. Webb, D. Tran, A. Davis, D. Pieri, R. LaHusen, J. Pallister, D. Dzurisin, S. Moran, and M. Lisowski. Optimized autonomous space in-situ sensor-web for volcano monitoring. *Aerospace Conference, 2008 IEEE*.

[19] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. In *OSDI*, November 2006.

[20] J. Zschau, M. Picozzi, C. Milkereit, K. Fleming, J. Fischer, F. Kuehnlenz, B. Lichtblau, and M. Erdik. The self-organising seismic early warning information network (sosewin). In *Eos Trans. AGU Fall Meet. Suppl*, number G43A-0668, 2008.