ROADNet: A network of SensorNets

Todd Hansen[†‡*] Sameer Tilak[‡] Steve Foley[†] Kent Lindquist[§] Frank Vernon[†]
Arcot Rajasekar[‡] John Orcutt[†]

[†] Cecil H. and Ida M. Green Institute of Geophysics and Planetary Physics
Scripps Institution of Oceanography
University of California, San Diego
La Jolla, CA 92093-0225 USA

[‡] San Diego Supercomputer Center
University of California, San Diego
La Jolla, CA 92093-0505 USA

[§] Lindquist Consulting, Inc.
59 College Rd. Suite 7
Fairbanks, AK 99701 USA

Abstract

As sensor networks become denser and more widely deployed, the potential develops for interconnecting these networks to combine datasets, share technological solutions, and to conduct cross-disciplinary research and monitoring operations that rely on several signal domains simultaneously. To that end, the Real-Time Observatories, Applications and Data Management Network (ROADNet) research project is connecting multiple sensor networks deployed by collaborating research projects into a single network in order to support a variety of research topics including coastal ocean observing, microclimatology and seismology. This paper gives a brief overview of the ROADNet project and discusses some of the implementation challenges we uncovered while building and maintaining the ROADNet system. We encountered challenges on several fronts including building effective programming abstractions for sensor networks, building tools for managing large-scale data in a scalable manner, and building efficient tools for deploying and managing hundreds of sensors. We discuss how these challenges were addressed and some of the lessons learned from collaborations with domain scientists using our network to conduct their research.

1 Introduction

The Real-Time Observatories, Applications and Data Management Network (ROADNet) research project ([1, 2]), `http://roadnet.ucsd.edu/`, is connecting multiple sensor networks deployed by collaborating research projects

*Corresponding Author: tshansen@ucsd.edu

Figure 1. An SDCOOS meteorological station, wireless communications and a wind generator installed on a lighthouse on Coronado Island, Mexico

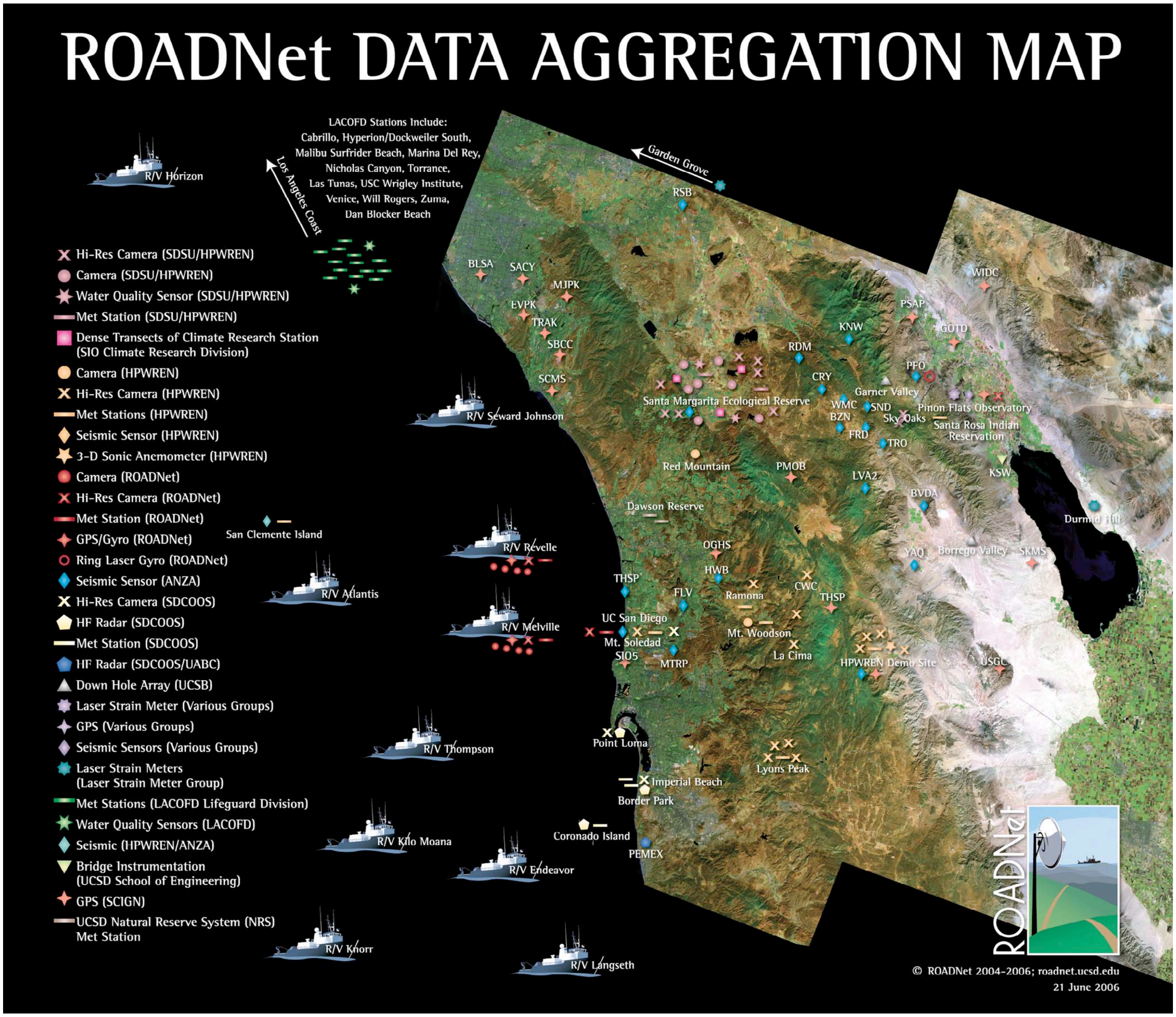

Figure 2. Instruments connected to ROADNet, the color denotes the collaborating group responsible for the instrument and the shape denotes the type of instrument deployed.

into a single network in order to support a variety of research topics including coastal ocean observing, microclimatology and seismology. This paper examines the difficulties, benefits, and obstacles encountered in combining heterogeneous networks. The combined network stretches from San Diego to Los Angeles (190 x 270 km) and connects 120 measurement stations of various types using multiple network technologies. Some key locations include two research vessels traveling throughout the world's oceans, seismic instrumentation along the San Jacinto fault, meteorological (met) stations and cameras to support fire fighting deployments, and region-specific, high density met deployments in support of specific research activities. The ROADNet project focuses on supporting new sensor types and studies sensor network integration issues. Figure 1 shows an installation by the San Diego Coastal Ocean Observing System (SDCOOS), `http://www.sdcoos.org/`, as part of its effort to instrument San Diego county's coast line with weather stations, cameras and surface current mapping HF radar. This station, on a remote island off Mexico, uses a wireless 802.11b network link to connect to the rest of our network. Figure 2 shows a map of our current sensor network. Since the data transport network is so complicated, we do not attempt to present these interconnections on the map.

Building and managing a large-scale sensor network is a daunting task. Heterogeneity in sensor hardware as well as data types and the need for high-availability of the network make the task even more challenging. We encountered challenges on several fronts; for example, building effective programming abstractions across multiple sensor networks and building efficient tools for deploying and managing hundreds of sensors. There are a number of important concepts that must be considered to support a large scale sensor network. The first involves the development of a scalable network fabric with common interfaces. This abstraction must translate across multiple platforms in order to accommodate ever evolving hardware. Other issues stem from dealing with heterogeneous data types since all of our sensors measure different things (e.g. meteorology, seismology, and coastal ocean observing); some don't even measure traditional time series data (e.g. imagery). In the next section, we provide a brief overview of the ROADNet project. In section 3, we discuss some of the lessons learned from our collaborations with domain scientists using the ROADNet network to conduct their research.

2 The ROADNet project

The ROADNet project is a collaboration between multiple research groups with the mission to develop an architecture that is scalable yet flexible enough to support heterogeneous sensor networks, robust data management and near-real-time data analysis. The ability to combine data from multiple research projects into a common system allows groups to leverage existing instrumentation while minimizing costs. This benefits researchers by increasing the number of sensors that can be used to study a particular problem, while limiting the amount of time spent on sensor maintenance and increasing the research productivity of each sensor deployed. Additionally, using a common back-end data storage grid allows researchers to concentrate on data analysis rather than data format conversion tasks. Integrating sensor networks into a single, large-scale, scalable system presents a broad array of challenges.

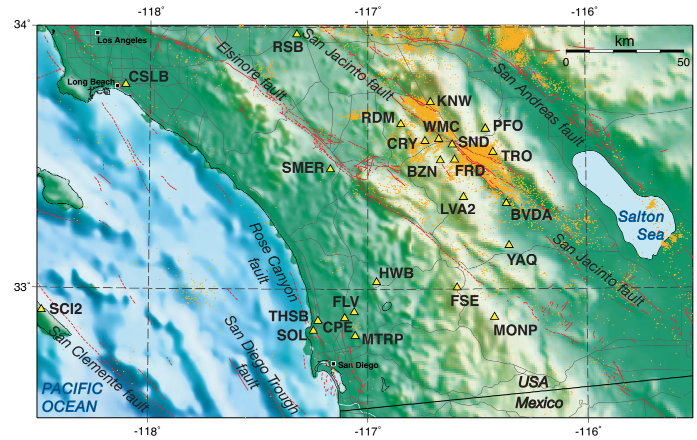

Figure 3. A map of the ANZA Seismic Network (yellow triangles) and earthquake epicenters (orange dots)

Each of the fourteen collaborating research projects has its own research objective and implementation plan. A couple of these collaborating projects are described here to give insight into their differing objectives. The SDCOOS project, `http://www.sdcoos.org/`, is focused on providing scientists, government agencies, and the public with near-real-time oceanographic, weather and water quality data for the San Diego coastal region. Such information is useful for tracking the flow of polluted water after a storm. The ANZA seismic network, `http://eqinfo.ucsd.edu/`, has deployed a real-time seismic network in the southernmost regions of California (Figure 3). The goal of this project is to provide on-scale digital recording of high-resolution three component seismic data for all earthquakes [3]. The data are made available in near-real-time to the California Integrated Seismic Network, other regional networks, the Advanced National Seismic System and to the general public in the San Diego region. The Los Angeles County Fire Department, Lifeguard Division (LACOFD) has created a network of web cameras, weather stations, and water thermometers, to aid in staffing beaches, tracking rescue activity and providing near-real-time information to the public and collaborating research projects. The LACOFD network provides a public website, `http://www.watchthewater.org/`, containing the results from their 72 mile coastline network. The following subsections focus on the implementation of our data network.

2.1 ROADNet Design Goals

The design goals of the ROADNet data network include providing a reliable near-real-time data network. To overcome this practical challenge, we use readily available off the shelf hardware and software systems such as TCP/IP and Boulder Real-Time Technologies', Inc. (BRTT) Antelope system [4].

Design Challenge: Reliable Data transfer over an unreliable network The first practical challenge that we encountered was providing users with near-real-time data access in the midst of unreliable networks. Our goal was to prevent unnecessary data loss due to network outages. We accomplished this by creating a buffered sensor network. In the event of a network outage, data will not be lost unless the outage exceeds the buffer's capacity (usually days to weeks). In order to provide the maximum reliability, we can push these buffers far out into the field near the sensors using embedded hardware.

Another of the ROADNet project's goals is to enable data sharing and research collaboration across multiple science domains. The goal of data sharing requires multiple layers of data abstraction to make data exchange and transport accessible between research groups and across multiple sensor types. ROADNet's solutions to this challenge are discussed below in Sections 2.2.1 and 2.2.4.

From our collaborations with domain scientists, we were able to learn more about user requirements for our sensor networks. As a result, this network also forms a test bed on which we can experiment with next generation networking technologies in an attempt to better address our collaborator's real-world network requirements. For example, the geographic extent of a normal sensor network is usually less than a couple of kilometers. To meet the research needs of our collaborators, we are operating this network over distances exceeding hundreds of kilometers.

2.2 ROADNet Infrastructure Practices

This section discusses current implementations and the standards with which our collaborators' networks need to comply. These implementation standards enable us to ensure that our infrastructure is scalable, yet adequately flexible to accommodate new sensors while providing an abstract interface to the data that limits the complexity visible to the end user. In our network, it is common for collaborators to deploy a data logger to convert their analog sensors into digital counts and buffer those digitized measurements for short periods until they can be downloaded. In the event of a network outage, this buffer enhances the ability to get data after the fact. For example, the Campbell Scientific CR23X data logger has 4 megabytes of internal memory (RAM) for storing data. Though data loggers often provide a data buffer, it is often difficult to retrieve data from the buffer reliably and the data buffers are susceptible to power failures. In contrast, the data buffers utilized by network nodes running Antelope software are automatically downloaded following a network outage and are not susceptible to power outages due to the fact that the data are buffered

ROADNet Software Layers

Abstraction	Layer	Used/Managed by	
5 abstract representations including timeseries and image data	Sensor Network Application Layers	Data Retrieval Apps Data Analysis Apps Data Archive Apps	Application Focused
Packetized Data Streams using various formats	Buffering and Sensor-to-Consumer data flow	Antelope Data Buffers	
Sockets containing multiplexed data packets	Transport Control Protocol (between data buffers)	Operating System	
IP packets	Network Medium (802.11b, cable modem, etc)	Hardware	Network Focused

Figure 4. ROADNet Sensor Network Protocol Stack

on disk. As a result, the maximum reliability is achieved by placing a sensor network node running Antelope as close as possible to the data logger. This minimizes the distance over which the interface is susceptible to failures resulting in data loss. In practice, about half of our data loggers have adjacent sensor network nodes. The other half utilize central sensor network nodes that connect to the data loggers over long distance Internet connections. This trade-off is discussed further in Section 3.3.

2.2.1 Infrastructure Challenge 1: Selecting the proper network fabric and data abstraction

Shortly after the initiation of the ROADNet project, it became clear that using non-standard network technologies or serial telemetry would severely limit the reach of our network and its ability to interface with multiple systems. Subsequently, we standardized on a TCP/IP stack with some additional levels of abstraction as shown in figure 4. The basic requirement of TCP/IP support provides an important abstraction that enables ROADNet to operate using existing Internet infrastructure. Additionally, using TCP allows us to relegate link-level errors to an established control layer. This allows us to concentrate on data abstraction (left most column) and sensor network data routing (second layer from the top). The data abstraction layer allows us to transport data through the buffer network independent of data type (images, time series, config files, etc) or format (compressed, raw packets, or etc). The result is that within the network we can use a representation that is most compatible with the data logger. This enables data logger specific compression and/or raw packet format preservation. Above the sensor network, these data are made accessible using abstraction libraries (libPkt and libuser). These C libraries are capable of taking data from a multitude of formats (as defined by the data logger interface developers) and providing a uniform representation. For example, Oregon Scientific weather stations and Campbell Scientific data loggers both send data encapsulated in formats that are specific to their data loggers. Applications accessing the data, however, simply use a standard time series representation to access all of the data. This allows us to use data logger specific packets that maximize compression or minimize software development time while we maintain the ability to use the same sensor network applications without modification. Two examples of these general applications include orb2db which records data to a long-term database and orbmonrtd which displays near-real-time time series data as soon as it arrives from upstream nodes. To make this work, it is simply a matter of providing a format definition to libuser.

In figure 4, the level above TCP/IP is the Buffering and Sensor-to-Consumer data flow layer. At the core of this layer is BRTT's Antelope system. Antelope creates a network of data buffers and data flow streams. While TCP provides reliable byte-stream delivery between hosts, the Antelope system ensures a reliable stream transport layer between sensor and data consumer (regardless of the number of buffer computers in the data path). Each stream may have many multiplexed data channels. A channel represents the time series data from one sensor at a particular station. Figure 5 shows an example data buffer network in which each sensor is assigned a stream name. This stream name is used to catalog information inside the data buffers. Transport applications use a regular expression (regex) to select which data are transported between buffers. In figure 5, the regex `/AZ_.*/` selects all data streams starting with `AZ_` to be placed in a database for long-term storage. As a result, the `AZ_TRO_SEIS` data stream will be recorded to the database. Effectively, network management becomes a task of providing buffer to buffer routing statements in the form of regular expressions. Current research activities involve methods to automate this routing so that applications can simply request data via regular expressions and the required intra-network routing will be implemented without user intervention.

As described above, the network abstraction provided by ROADNet allows us to operate across any TCP/IP capable network. In implementation, we use 802.11b wireless links, proprietary wireless links, consumer grade cable modems, satellite modems, and even government WANs and LANs.

2.2.2 Infrastructure Challenge 2: Standardizing Node Hardware

One of the overriding concerns when selecting hardware is that it changes rapidly. For example, a product purchased for the past two years may suddenly become unavailable. To minimize the impact of rapidly-evolving hardware, ROADNet chose a software package that operates on a variety of hardware platforms. This principle allows us to support embedded platforms such as X-Scale and x86 processors running Linux, while at the same time supporting Sun servers running Solaris and user-friendly MacOS X work stations. In order to support such a variety of hardware, we require hardware that is substantial enough to support ANSI C compilers and a POSIX compliant operating system. Table 1 shows the combinations of hardware and software that we currently support.

Processor	Operating System
Strong-Arm & Compatible (Intel X-scale)	Linux
x86	Linux
PowerPC	MacOS X
Sparc	Solaris

Table 1. Processor / Operating System combinations supported by ROADNet

In terms of embedded processors, we chose to limit ourselves to devices that can support a full TCP stack and a standard C-Library implementation (i.e. GNU's glibc). Experience has shown that project engineers can spend numerous hours modifying their systems to operate on a particular hardware solution, only to have the cost of such an investment outweigh the cost of having selected more robust hardware in the first place. Table 1 shows that we support Strong-Arm processors running the Linux operating system. This does not mean that we only support one piece of hardware. In less than five years, we have operated our sensor network software on embedded hardware from four separate vendors using three different Strong-Arm compatible processors. Our current embedded system deployments involve only one of these vendors, Arcom's Viper, shown in figure 6. However, our installed network comprises hardware from all of the vendors. By committing to a minimum hardware standard, we were able to diversify the hardware types supported. This enables us to integrate different types of users and to successfully make the transition from one hardware implementation to another when a better product is available or an older product becomes obsolete.

2.2.3 Node Software

ROADNet requires an operating system (OS) which supports standard C-library calls, is POSIX compliant, supports threads and has a well tested TCP/IP implementation using

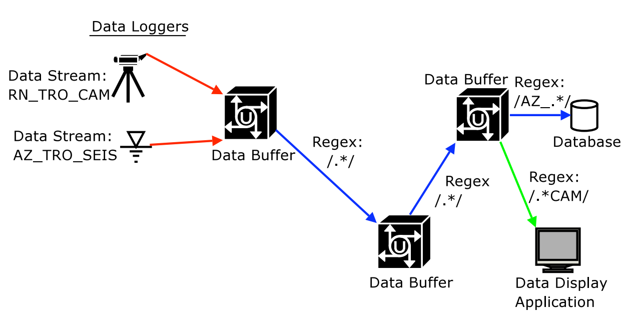

Figure 5. Example Sensor Network

Figure 6. Arcom's Viper Embedded Computer

Berkeley-style sockets. At this time, our system is running on the following operating systems: Linux, Solaris and MacOS X. We specifically do not support systems using uClibc or tinyOS because they require slightly different interface code than traditional libc systems. However, the embedded systems we use easily support a full glibc installation and most of them can even support native compilers. This greatly reduces the time it takes to support new hardware.

Once the OS is installed on a node, we then install BRTT's Antelope system. This system was originally developed for transporting seismic data between stations, analysts and storage facilities. It forms the data buffer and transport backbone of the ROADNet system.

The Antelope system provides an object ring buffer (ORB) that can be used to buffer data on each sensor node. It stores data in a ring buffer where the oldest data are overwritten by the newest data. Data transport applications connect to this temporal buffer and transfer data. During a network outage the temporal buffer fills, but the data transport application cannot download the data. Once network access has been restored, the data transport application connects to the buffer and reconnects to the location where it last retrieved data. This provides a reliable data transport system that can cope with short to medium length outages. Data may still be lost if the outage is longer than the temporal life span of the data buffer. In most of our systems, that life span is on the order of a week or more.

Once data are received at a location where it may be useful for analysis, a researcher can either access the data directly from the local buffer or have another application write the data to a Datascope database for long-term storage [4].

2.2.4 Infrastructure Challenge 3: Supporting heterogeneous data and sensor types

The abstraction provided by our data management system allows us to support a large variety of sensor types. However, writing software and developing hardware to support a large variety of individual sensors is an obstacle to the implementation of any data network covering multiple scientific disciplines. In order to tackle this problem in a scalable fashion, we looked for a sensor to network interface that provided the appropriate level of connectivity and reliability between sensors and the sensor network fabric while providing for the largest variety of sensors and data types. At the same time we had to find a middle ground that would minimize time spent writing interface software and provide a clear demarcation between the sensor and the sensor network so that debugging data outages would be straightforward. In our experience, there is no point in writing software and engineering hardware to sample individual sensors unless enough will be deployed to recoup all of the development costs. Since most projects are not large enough to benefit from such tasks and our network includes a number of diverse projects using different sensor types, we focused on interfacing with off-the-shelf data loggers, such as the Davis Instruments weather station. The ROADNet project has written open source software to support 17 varieties of data loggers sampling everything from geophysical data to ocean surface currents to meteorological data. In addition, BRTT's Antelope software provides support for a number of seismic data loggers.

In addition to providing conversion from analog sensors to digital counts, a vast majority of data loggers also maintain temporary buffers of recent data samples. This allows for easy recovery of data in the event of a brief outage. In addition, off-the-shelf data loggers are separate pieces of hardware; in the event of a failure, the device can be tested using software provided by the manufacturer and shipped back for repair. Each data logger can provide multiple channels of data from multiple types of sensors located at a single station. By interfacing with a single data logger we are able to collect data from a variety of sensor types. We name each of these data channels using the following scheme:
`SensorNetwork_StationName_ChannelName`
This naming convention, developed by BRTT, provides a classification hierarchy, by which data are cataloged throughout the ROADNet system (i.e., `SDC_CI_AIRTEMP`). A researcher can navigate down the hierarchy looking at all of the data available from the SDCOOS project or they can traverse the hierarchy looking for all stations reporting air temperature data.

Once the data are named and injected into the ROADNet system, they become accessible through the standard Antelope API, independent of where they came from or what they measure. As a result, data consumers are no longer required to write data logger specific interfaces or to reformat the data from different loggers in order to compare the results.

By limiting our sensor interface to the data logger level, we are able to support off-the-shelf products and contain the complexity in a data logger interface program that is written once and available to all. Following acquisition, our software system treats each sensor as a separate data channel and provides a standard API by which all of the sensor network data can be accessed.

The same is true for data that do not fit the traditional one-dimensional time series model. There are only a handful of data representations that are visible to the end user and the end user is only required to support the data representation that fits the type of data they are collecting. For example, if a user is only interested in one-dimensional time series data, then she must only implement support for a single data representation. Later, if that same user wishes to support real-time camera images, then the user will need to extend her software to support this second representation. We make every attempt to minimize the number of data representations that are required to access ROADNet data. Aside from camera data, 90% of our data are accessible via a single time series representation.

2.2.5 Infrastructure Challenge 4: Scalable management of large-scale sensor data

We are currently collecting data from 55 different stations comprising 246 sensors. Sixty-five additional stations provide data that do not resemble traditional one-dimensional time series data. For example, some stations have cameras that capture multi-megapixel images every couple of minutes. These data can be really useful, but no one is capable of examining data in real-time 24 hours a day. In addition, analysts might sometimes need to review older data in order to determine if a new event is distinct. In order to solve these practical challenges, we operate a network of data repositories. We run one repository at the Scripps Institution of Oceanography that stores all of the data collected by our project. In addition, most of our collaborators run repositories to store data from sensors of interest to their research.

To enable these repositories, we use the Antelope data buffering and transport system to transport data from the data logger interface code to Datascope databases on each repository. This provides us with nominally near-real-time databases. Outages, of course, can cause data to be delayed. In addition, repositories may use the Storage Resource Broker (SRB) [5] to maintain an off-site deep archive of the data. This is useful for offloading older data that would exceed the local disk space and for backing up recent data so that they will not be lost in the event of a system failure. The combination of data backup using the SRB, the central ROADNet repository and our collaborators' repositories provides us with as many as three replicated copies of our data, in multiple physical locations.

2.3 Sensor Network Management Challenges

Managing a diverse sensor network can be a rather difficult task. The ROADNet project uses a number of tools in order to make the problem more tractable.

2.3.1 Network Management Challenge 1: Status and Notification

Since it is impossible to implement a large scale sensor network without deploying a significant amount of computing hardware, it is important to know if these devices are operating correctly and to know when a particular feature stops working. This presents an implementation challenge that all sensor networks must address.

To determine the status of both network infrastructure, sensors, and collected data, ROADNet uses a wide variety of software checks executed by a monitoring server. These checks vary from simple ping and TCP port checks, to SNMP checks, to latency and value checks of sensor data at multiple buffer locations throughout the network. These tests are scheduled, executed, and displayed using the Nagios® service and network monitoring program (`http://www.nagios.org/`) and an alert is sent to the relevant operator of hardware which is not in a normal operational state. An alternative to Nagios® would be to create a ping script to periodically email system operators when a host becomes unreachable, but this does not provide the ability to implement other service checks (e.g. is data from sensor X arriving?), to track service history, or to manage technician comments.

2.3.2 Network Management Challenge 2: Visualization of sensor network topology

In order to support ROADNet's network of sensor networks (Figure 2), we have deployed a large network of data buffer/access nodes. Managing this topology can be difficult. One way to alleviate some of this burden is to ask the nodes to generate a catalog of what they are doing, what data they have and with whom they are speaking [6]. We have developed a software package (pforbstat/orbtopo) that manages this task and displays the results. We simply run an instance of pforbstat on each node to generate a local catalog. The catalog flows through the network as if it were sensor data. At the data repositories, we run a simple software application, orbtopo, that extracts the individual catalogs and generates a "click-able" topology map. Orbtopo can highlight orbs containing data matching a regular expression or we can drill down to find out which sensor interface applications are running on a particular orb. In summary, automated topology mapping provides a way to visualize what is actually running, not what you believe is configured. Interested readers may navigate ROADNet's current topology interactively at `http://mercali.ucsd.edu/orbtopo.cgi`.

An alternative to our sensor network visualization tool (orbtopo) would have been to use the Cooperative Association for Internet Data Analysis' Otter visualization tool [7]. However, the tool does not provide the topology analysis portion of the task and it does not provide a method for drilling down into the map.

2.3.3 Network Management Challenge 3: Software Configuration, Deployment and Maintenance

Despite a variety of hardware and software platforms for buffer nodes scattered across the network, software and configuration updates must still take place. To address this challenge, ROADNet uses Cfengine (`http://www.cfengine.org/`) to maintain a central repository of configuration settings and software packages. Nodes check in with the configuration server on a frequent basis, download their configuration, and then perform a self-check to see if they need to make any corrections to their configuration. Using this system allows us to group hosts so that software and configuration modifications can be deployed and maintained to a large number of machines with very little effort.

An alternative to Cfengine is rsync. However, rsync does not support service management. For example, if you wanted to update the configuration of a web server, you would need to restart the web server after the changes were complete. Cfengine provides facilities to automate these configuration management tasks.

3 Lessons learned from user experiences

In addition to the practical challenges discussed above, we have learned a number of sensor network infrastructure lessons while working with various collaborators as they build out their sensor networks.

3.1 Selection of Dataloggers

Collaborators typically select data loggers based on the needs of their research. As such, there is a wide variety of data loggers that we must support. These include off-the-shelf units and home grown data loggers. This unregulated selection of data loggers presents a practical challenge. In theory, the IEEE 1451 sensor interface standard will provide a solution to this problem. At this time, however, there are no deployable 1451 sensors that meet the needs of our collaborators.

It has been our experience that preferentially supporting off the shelf data loggers allows us to concentrate our software development efforts. For example, if the SDCOOS project would like to use a data logger that we haven't encountered before, there is a larger chance that another group will also want to use that same data logger if it is a commercially available product. As a result, the time spent developing support for an off-the-shelf implementation is more beneficial since the data logger can be used by other research groups as well. A home grown data logger does not have that immediate potential since a lot of groups are hesitant to purchase another group's hardware without assurances of long-term product support and availability.

In ROADNet's operational network, we support a large variety of data loggers in order to provide flexibility to our collaborators (some data loggers are inherently better at certain tasks than others). However, we have observed that a large number of stations use the same data logger for the same purpose even if they are run by different collaborators. This has grown to the level where occasionally new collaborators will ask for advice on selecting data loggers before they deploy infrastructure. As additional collaborators adopt a particular data logger for their needs, our time invested integrating that data logger into our network becomes more beneficial. On the other hand, we do still support users of home grown data loggers as long as there is a solid reason. We currently have three home grown data logger types in use throughout our network.

3.2 Costs of building your own datalogger

There are a number of groups that would like to deploy their own custom data loggers. Most of the time it is to reduce costs or power consumption. Custom data loggers tend to implement non-standard data interfaces and are often only deployed by the research project which designs them. This data logger interface uniqueness presents an implementation challenge for interconnecting their system with ours, since integrating a different custom data logger into our network infrastructure for each new collaborator is not scalable. In general, most of these users tend to forget that human time spent developing a new data logger is often more expensive than simply using more expensive hardware in the first place. In addition, these special purpose one-off solutions tend not to be maintainable in the long term since a critical mass of users is not realized.

For example, we had one collaborator decide to deploy a dense array of high quality measurement stations using their own hardware to reduce the per unit cost. To further reduce costs, they chose to implement their network using batteries so they were constrained by power. The engineering time spent building a system to manage these complexities over a 3x3 km area accumulated to multiple man-years. For that cost, they could have deployed off-the-shelf data loggers and solar panels. At one point, the engineering time spent dealing with implementing a communication strategy capable of supporting the low power requirement alone exceeded the cost of providing sufficient solar power for each station.

For future collaborators deploying new infrastructure, ROADNet engineers have found it worth their effort to review the total system costs involved in using a home grown data logger. In some situations, it is desirable to invest the extra effort to integrate a custom data logger, however in most situations there is an off-the-shelf solution that will save both time and money.

3.3 The quantity of sensors deployed versus the reliability of individual sensors

A number of collaborators are initially interested in deploying a highly-reliable network of sensors. However, due to the long duration of our sensor network deployments, it is likely that some hardware will fail. Providing more reliable infrastructure to accommodate those failures comes at a cost. For example, one of our collaborators is deploying a meteorological network using increased reliability techniques (a data logger and data buffer colocated to ensure zero data loss in the event of a network outage). This costs them twice as much as the original data logger. If they had used an unbuffered serial to ethernet converter in place of a data buffer, 33 stations could have been deployed for the same cost of 20 in the original configuration. This significantly increases the density of their network, perhaps allowing them to interpolate data when a station is offline. In addition, the data loggers they chose are less reliable than their network, so spending money making a reliable network infrastructure may not be as beneficial as it sounds. As a result, in the future this collaborator is planning to assign each data buffer to manage the data from 2 or 3 nearby data loggers.

3.4 Balancing power consumption versus hardware capability versus staff time

In the world of sensor networks there is a valid concern for managing power consumption. This is even more important as sensors become small and expendable. Due to the geographic expanse covered by our network and the inter-station spacing, we have chosen to focus on stations that can be deployed near available power sources or easily powered by solar power. In most cases, the radios to connect stations to a network access point 20 km away consume more power than the stations themselves. In addition, the cost of each station is too high for them to be disposed of when a battery wears out and the time to drive to these stations to replace batteries is not scalable in a widespread network. A specific example is the case of a seismic station comprised of a broadband velocity sensor, a strong motion accelerometer, a datalogger, and a radio. These stations are commonly deployed at distances from 10 to 100+ km away from a telemetry connection point. The sensors and dataloggers combined power consumption is two watts. However, to power the continuously operating 9.6 kbps telemetry link up to 80 km requires radios consuming 5 watts. The ROADNet project has focused its efforts on deploying infrastructure that is self sufficient in terms of power. As a result, a lot of our infrastructure is connected to AC-power. That means that power consumption is for the most part not an immediate concern for our collaborators.

Where it is a concern, there are a number of options available for southern California including solar power and wind generators. An example of this is the SDCOOS installation on Coronado Island, figure 1. For this installation, SDCOOS has deployed a solar panel array, a wind generator and deep cycle batteries in order to provide continuous power capable of supporting their HF radar system, a met station, and two wireless telemetry links. A less power hungry example is the Santa Rosa indian reservation's solar relay weather station. This station was added after an existing solar powered Wi-Fi relay was already in place. The addition of a weather station did not significantly increase the power budget of the station. On the other hand, for collaborators who are routinely visiting stations to do periodic calibrations, replacing a battery may be only an incremental addition to their work load.

3.5 Sensor network routing management

At this time, data transfer from data buffer to data buffer is configured by hand using regular expressions. We have found this to become unwieldy for large networks. As the network becomes more complex the potential for operator error increases. Using the self-cataloging nature of pforbstat (see Section 2.3.2) and the fact that each application registers a regular expression (regex) to select the data in which it is interested, we are developing an algorithm to route near-real-time data to the consuming application automatically. This form of automatic data routing allows users to find, connect, and analyze data contributed by multiple collaborating research projects in near-real-time. This software is currently in the testing phase; a paper will be submitted in the near future.

4 Discussion

Large-scale environmental observing systems are poised to become the dominant means for studying a variety of natural phenomena. EarthScope [8], NEON [9], and ORION [10] are examples of such large-scale observatories which will comprise thousands of sensors, tens of thousands of data streams, and many end users. We believe that the lessons learned from the ROADNet project will be directly applicable to these systems.

5 Acknowledgments

This work and the ROADNet project are sponsored by grants from the National Science Foundation (OCE-0121726 and ATM 03-25963), the Office of Naval Research (N00014-98-1-0772), and with matching funds from the California Institute of Telecommunications and Information Technology, Scripps Institution of Oceanography, San Diego State University, and the Cecil H. and Ida M. Green Institute of Geophysics and Planetary Physics. We thank Jennifer Matthews and Rob Newman for their assistance with figures.

References

[1] F Vernon, T Hansen, K Lindquist, B Ludäscher, J Orcutt, A Rajasekar; *ROADNET: A Real-time Data Aware System for Earth, Oceanographic, and Environmental Applications*, Eos Trans. AGU 84, Fall Meet. Suppl., Abstract U21A-06

[2] H-W Braun, T Hansen, B Ludäscher, J Orcutt, A Rajasekar, F Vernon; *Distributed Data Management Architecture for Embedded Computing*, HPEC 2002

[3] Vernon III, F.L. (1989). *Analysis of data recorded on the ANZA seismic network*, Ph. D. Thesis, U.C. San Diego

[4] BRTT's Antelope Real-Time System
`http://www.brtt.com/`

[5] SDSC's Storage Resource Broker Project
`http://www.sdsc.edu/srb/`

[6] K.G. Lindquist, T.S. Hansen, B. Ludäscher, F.L. Vernon; *Dynamic Dataflow Topology Monitoring for Real-time Seismic Networks*, Eos Trans. AGU 84, Fall Meet. Suppl., Abstract ED32C-1205

[7] B. Huffaker, E. Nemeth, K. Claffy; *Otter: A general-purpose network visualization tool*, ISOC Inet99.

[8] The EarthScope project
`http://www.earthscope.org/`

[9] The National Ecological Observatory Network
`http://www.neoninc.org/`

[10] The Ocean Research Interactive Observatory Networks
`http://www.orionprogram.org/`