

Sensor Networks and Remote Sensing Processing
Algorithms for Hydrological Applications

Michael A. Murphy
Departmental Honors Thesis
Department of Computer Science

Dr. Christopher J. Post, Advisor
Department of Forestry and Natural Resources

Clemson University

April 28, 2005

Abstract

Sensor networks and remote sensing devices are revolutionizing the way environmental data is obtained, providing better real-time and logged data for environmental research and operations. Two separate but related projects have been undertaken in order to extract data for hydrological applications within the overall environmental domain. The first of these projects is a piece of computer software that provides estimates of total precipitation accumulation within a single watershed. In order to produce this estimation, the program extracts data from a national hydrological product called a Stage IV analysis. These data are then analyzed within a Geographic Information Systems (GIS) context to produce a single estimate of watershed precipitation. On the ground, the effects of this precipitation can be measured by the second project: a set of sensor networks for soil moisture and stream turbidity monitoring. These networks, comprised of small devices called motes, provide the ability to monitor actual observed events in the field. Together, these sensor networks and remote sensing devices provide valuable information for analyses of hydrological phenomena and land-use changes.

Contents

1	Introduction	3
1.1	The Problem Domain	3
1.2	Water Resource Change Assessment	4
1.3	Technological Issues	5
1.4	Overview of the Research	5
2	Watershed Precipitation Estimation	7
2.1	Overview of the Estimation Algorithm	7
2.2	The Stage IV Precipitation Product	8
2.2.1	Description of the Stage IV Product	8
2.2.2	Relationship between the Stage IV product and the Doppler Weather Radar	8
2.3	Data Preprocessing	9
2.4	The Numerical Integration Algorithm	10
2.4.1	Surface Integration as a Concept	10
2.4.2	Riemann Sum Procedure	11
2.4.3	Differential Area Calculation	11
2.5	Optimization for Uniprocessor Platforms and Time Complexity . .	12
2.6	Testing the Implementation	13
2.7	Sources of Error	19
2.8	Conclusions and Potential Future Changes	20
3	Soil Moisture Sensor Network	21
3.1	Overview of the Soil Moisture Network	21
3.2	The Software Design	22
3.2.1	Nondeterministic Finite State Machine Model	22
3.2.2	nesC and TinyOS Programming	24
3.3	The Experimental Method	24

3.4	Results of the Deployment	25
3.5	Conclusions and Changes for the Next Deployment	27
4	Turbidity Sensor Network	29
4.1	Introduction to Turbidity Sensing	29
4.2	Hardware Design	30
4.3	Software Overview	32
	4.3.1 Distributed Logging	33
	4.3.2 Network and Gateway Communications	33
4.4	Deployment Plans	34
5	Conclusions and Future Research Topics	35
5.1	Research Conclusions	35
5.2	Future Research	36
6	Epilogue	38

Chapter 1

Introduction

1.1 The Problem Domain

Infrastructure development has profound impacts upon the environment. Changes in land use affect the natural balance of the ecosystem, which in turn affects the population relying on natural resources. This phenomenon is especially noticeable with water resources, as deforestation and other changes along and near rivers, lakes, and streams can have substantial impacts on the quality and quantity of water within the systems. Issues such as these impact not only the potable water supply, but also the flora and fauna that rely upon the water source for subsistence.

Two human areas greatly impacted by water quality and quantity issues are potable water supply systems and hydrological control structures such as dams and dikes. Excess sediment washing into watersheds can be transported downstream into water supply reservoirs, necessitating the use of additional filtering and purification mechanisms for providing healthy drinking water. Excessive runoff from heavy rain events can push rivers and streams above flood stage, inundating agricultural lands and destroying crops, structures, and transportation systems. In extreme cases, excess water also can lead to sudden structural failures of dams and dikes, resulting in catastrophic damage to nearby areas.

In addition to the effects of runoff and sedimentation, there has been increasing concern for the potential of deliberate tampering with potable water supplies and impoundment structures. Following the terrorist attacks of 11 September 2001, there has been concern among various government bodies that terrorists might

attempt to poison water supplies with harmful substances. Also, there is the potential for attacks on impoundment structures, the effects of which could result in catastrophic flooding and widespread loss of life [United States Environmental Protection Agency, 2004]. Because of the vital role water plays in human life, natural and man-made threats to hydrological resources pose great risks for society.

1.2 Water Resource Change Assessment

Traditionally, water resources have been monitored by isolated, in-situ devices located sporadically at points along the watershed. Many of these devices are river gages, tipping-bucket rain gages located along the water course, and isolated water quality sensors. Most of the time, these sensing devices are located at considerable distances from each other, meaning that changes along the watershed cannot be accurately pinpointed. Depending upon the instrumentation, it is also conceivable that some changes might go unnoticed unless and until the change propagates to a downstream sensor. In the case of watersheds in and upstream of potable water supply areas, these changes could prove quite troublesome.

To remedy the general lack of detailed watershed information, it will be necessary to increase the spatial density of sensing devices. However, many important watersheds are located in areas in which grid power and data communications are not readily available. It is therefore necessary to obtain data in a power-efficient manner, in such a way that extremely low-bandwidth communications (e.g. meteor-burst) can be used to relay information back to the scientists. One solution to this problem is to forgo in-situ sensors in favor of remote sensing devices, such as radar and satellite. However, the errors associated with remote sensing can be substantial, and the information processing required to extract useful information from these devices can be significant, as is seen in Doppler radar precipitation estimates [Ulbrich and Lee, 1999].

Since the ultimate solution for providing information is to use more in-situ sensors, sensor networks are an attractive option. These networks consist of low-power nodes equipped with wireless communications and logging capabilities, which are ideal for deploying several sensors in remote areas where battery and solar power are the only available energy sources. Though it is anticipated that these types of networks will become ubiquitous in the future [Ricadela, 2005], remote sensing

devices will continue to be needed for assessing overall regional trends, such as precipitation accumulation. Additionally, remote sensing devices will continue to be the only means by which to estimate conditions in some extremely remote areas where deployment of in-situ sensors is not possible or feasible.

1.3 Technological Issues

Issues with sensor network and remote sensing technologies exist in two places: the sensor devices themselves and the processing used to extract useful information from those devices. With sensor networks, these issues are primarily resource constraints, particularly in terms of energy and processing resources. In the case of remote sensing devices, the primary issue lies with signal processing of received input, which is generally electromagnetic radiation (light, infrared, or radio waves). On the data processing side of either system, the primary issue arises in terms of the sheer size of the computational problem. Since these sensory data are usually geospatially referenced in a Geographic Information Systems (GIS) context, processing is almost always $O(N^2)$ or worse due to the fact that there are at least 3 dimensions involved (the 2-dimensional geospatial reference and at least one dimension of data). Substantial algorithm optimizations, along with high-end hardware, are needed to process these data in an efficient manner.

An additional issue with developing software for sensor networks is the complexity of the software design and implementation process. The *de facto* standard software and hardware combination for sensor network development is Berkeley TinyOS running on the Berkeley Mote. TinyOS applications are written in nesC, a highly granular, object-oriented C dialect. Due to the complex granular structure of the language, along with missing, incorrect, and outdated documentation, programming in this environment is time-consuming and error-prone.

1.4 Overview of the Research

Several distinct, and yet related, experiments have been undertaken to assess hydrological parameters. These projects consist of one algorithm for estimating accumulated precipitation from Doppler radar and surface rain gage data, one sensor network for soil moisture measurement, and one sensor network for stream turbidity (suspended sediment) monitoring. The radar accumulated precipitation estimation

algorithm provides a total estimate of the amount of water that accumulates in a watershed during a rainfall event. For monitoring the paths taken by that accumulating water, the soil moisture sensor network detects ephemeral streams that form whenever precipitation occurs. With knowledge of the location of these temporary runoff channels, it is possible to assess how much sediment will be washed into the stream. The second sensor network (still in development) provides a confirmation of this assessment, along with an analysis of sediment introduced upstream, by measuring the amount of turbidity present in the water.

The next three parts of this paper explain the implementations, testing, and results of each of the projects. Following these detailed sections, the overall research outcomes are discussed, and possibilities for future research are identified.

Chapter 2

Watershed Precipitation Estimation

2.1 Overview of the Estimation Algorithm

When precipitation occurs over a watershed, it is difficult to determine the total amount of precipitation that has fallen because precipitation amounts often vary from location to location, and surface rain gages are deployed in a relatively sparse manner. As a result, estimates of total rainfall taken from gages alone tend to underestimate the total amount of precipitation within the watershed. Without accurate information about the observed precipitation, it is difficult to make predictions about stream water quantity and quality. Such predictions can be improved if more information is available on the nature and amount of the precipitation over the watershed.

Reed and Maidment [1995] developed a set of programs to obtain precipitation and average flow length from National Weather Service (NWS) Stage III precipitation data, using Geographic Information Systems (GIS) software. Since that time, the National Centers for Environmental Prediction has been organized into a single body, and a newer Stage IV analysis with manual quality-checking is produced. As the emphasis on this study is in calculating a precipitation total in order to make water quality predictions, it was determined by analysis that a numerical approach to the problem would yield more applicable results than a GIS-produced method, because it is more efficient and easier to automate.

The general numerical approach taken in this project was to perform an integral in three dimensions, in order to produce a single quantification of the total precipitation accumulation from the Stage IV product. In order to implement this procedure computationally, a three-dimensional Riemann sum was used as the integration method, with the two-dimensional boundaries of the watershed given as a polygon. The topographic elevation dimension was not considered, as the boundaries of the watershed to be studied are well known.

2.2 The Stage IV Precipitation Product

2.2.1 Description of the Stage IV Product

Stage IV precipitation accumulation products, which are prepared and quality checked several times daily by the National Weather Service's River Forecast Centers (RFCs), are made available via anonymous FTP servers by the National Centers for Environmental Prediction (NCEP). This product is generated by combining data from the National Weather Service Doppler radar system and surface rain gages, and it has a nominal resolution of approximately 4 km by 4 km [Lin, 2002]. Data points are encoded in Gridded Binary (GRIB) and stored in a north polar stereographic map projection. Data points within the Stage IV product are single-precision floating-point values that correspond to the estimated precipitation for the associated grid cell, with units of $\text{kg} \cdot \text{m}^{-2}$.

One important attribute of the Stage IV analyses is that they are checked by both machine algorithms and NCEP staff prior to release. Thus, the errors in estimation that could occur when using direct Doppler radar products, such as the level 3 estimated precipitation product, should be somewhat fewer in number. It is also important to note that while Doppler weather radar data forms a major component of the Stage IV product, this analysis is not an actual product of the Doppler radar system.

2.2.2 Relationship between the Stage IV product and the Doppler Weather Radar

Although the Stage IV precipitation product is derived in part from the National Weather Service Weather Surveillance Radar, 1988 Doppler (WSR-88D, also called NEXRAD) system, the Stage IV product is not generated directly by the WSR-88D.

It is easy to confuse the NCEP Stage II and IV analysis products with WSR-88D Level 2 and 3 radar products; however, these product categories are fundamentally different.

The WSR-88D radar system uses four “levels” of product data, which do not correlate in any way to the NCEP “stages.” In the radar system, level 1 data comes from the radar antenna, where it goes to the Radar Data Acquisition processor. That processor, abbreviated as the RDA, produces level 2 data that consists of identified points, intensities, and Doppler phase shift information. From there, the data are further processed by the Radar Product Generator (RPG) into binary graphic files displaying reflectivity, velocity, rainfall estimated accumulation, and other parameters. These products are called “level 3” products, and imagery or data derived from them is termed “level 4.” [NOAA, 1992]

In contrast, the NCEP stage data undergoes more processing. The WSR-88D level 3 precipitation analyses are combined with data from surface rain gages on NWS computer systems. This combination of data sources is done for each River Forecast Center region, but a mosaic of all the RFC regional products is released by NCEP as the “Stage II” analysis. Meanwhile, staff at each River Forecast Center manually quality-check their regional multi-sensor analysis product, in order to attempt to rectify estimation errors. These quality-checked regional products are mosaicked into a single national product at NCEP, and this is the product that is released as the Stage IV analysis. [Lin, 2002]

2.3 Data Preprocessing

Because of the complexity of processing GRIB files directly, NCEP’s free wgrib decoder was employed [Ebisuzaki, 2004]. This decoder has the capability to output the grid size, followed by a text representation of the value at each grid cell, using lines to separate the values. In order to make it easier to access the data from the grid using the row-major C programming language, the first task was to write a simple program to change the formatting to rows of columns, with spaces between column data and carriage returns between rows. This intermediate format was termed “text grid from GRIB,” or TGG.

From the TGG intermediate file, the data were loaded into memory and a 3-dimensional Riemann sum was used to perform a surface integral on the data, with the region of integration bounded by a polygon describing the watershed. For ease of polygon input, the standard ESRI Shapefile format was used (ESRI, Redlands, CA USA) via the open-source Shapelib library [Warmerdam, 2003]. As this tool was designed to be run on only one watershed, the executable program was implemented such that it integrates on only the first polygon in the shapefile. In order to extract the value from the TGG record, a FORTRAN-77 routine (provided by NCEP) was used to convert latitude and longitude to grid coordinates [Lin, 2002]. By adjusting the call to this routine, the coordinates were switched from the column-major FORTRAN to the row-major C. Following the integration procedure, the program placed one numerical value onto the standard output stream: the total amount of precipitation, in liters, that fell within the polygon.

2.4 The Numerical Integration Algorithm

2.4.1 Surface Integration as a Concept

The first step in calculating the precipitation total was to understand that the process of summation is merely a surface integral. At any point inside a region bounded by a polygon, the discrete value of the Stage IV rainfall estimate can be expressed as a function of the coordinates of that point. To simplify the mathematics, latitude and longitude were used as the coordinate system for the polygon points, because in concept latitude is orthogonal to longitude at every point on the earth's surface. Conceptualizing the Stage IV value as an altitude above the polygonal surface, the mathematics were kept in a three-dimensional vector space with an orthogonal basis.

Given the orthogonal properties of the vector space, the surface integral was reduced to a simple double integral over the polygonal region R:

$$\iint_R (f(\phi, \lambda) \cdot g(\phi) \cdot h(\lambda)) d\phi d\lambda$$

The raw value of the Stage IV estimated precipitation accumulation (in $\text{kg} \cdot \text{m}^{-2}$) is given by the function f at the specified latitude and longitude points. Functions g and h are geospatial translations from latitude and longitude coordinates to meters. The resulting integral value has the units of $\text{kg} \cdot \text{m}^{-2} \cdot \text{m}^2$, or kilograms. Since a mass of 1 kg of water occupies a volume of 1 L under standard atmospheric conditions, the units are taken to be liters.

2.4.2 Riemann Sum Procedure

For computational simplicity, a Riemann sum numerical integration technique was adapted for use in three-dimensional space. Rectangles were used as the approximating shape, subdividing the bounding box of the polygonal region according to a step size in degrees of latitude and longitude. From the starting coordinate of this bounding box, each rectangle was checked to see if it was within the polygon. The value of the grid under the rectangle was multiplied by the size of the rectangle to calculate the volume of rainfall under that rectangle. For each rectangle determined to be inside the polygon, the local accumulation amount was added to the overall total. As the step size, or precision, of the rectangular subintervals approached zero, the value of the Riemann sum procedure approached the true value of the integral.

2.4.3 Differential Area Calculation

While the great-circle distance formula is a popular means of arc length measure, it was determined that the round-off error present in that formula was significant for measuring points that are relatively close together, due to the effects of calculating trigonometric functions within the constraints of double-precision floating-point values. Thus, a differential area calculation was derived from an extremely simple geometric projection. A function was designed to determine the area (A) completely from the change in latitude ($\Delta\phi$) and longitude ($\Delta\lambda$) over a single rectangular subinterval. Given a spherical approximation of the earth's radius, R , and average latitude over the subinterval ($\bar{\phi}$), the equation was derived to be:

$$A = R^2 \cdot \Delta\phi \cdot \Delta\lambda \cdot \left(1 - \frac{2\bar{\phi}}{\pi}\right)$$

Table 2.1: Execution Time Versus Precision

Precision (powers of 10)	Non-Optimized	Optimized
-3	0.03 s	0.764 s
-4	3 s	0.751 s
-5	300 s (5 minutes)	0.757 s
-6	30,000 s (over 8 hours)	0.798 s
-7	(not tested)	1.231 s
-8	(not tested)	5.785 s
-9	(not tested)	50 s
-10	(not tested)	505 s (8.4 minutes)

Precision, in powers of 10, represents 10^p , where p is the precision level (e.g. a precision of -3 means 10^{-3} , or 0.001), or the size of the step taken upon iteration (in degrees of latitude and longitude). The program running times in seconds, for the Reedy-Saluda watershed, are given for both the unoptimized and optimized versions of the program.

The net effect of this projection is that distances corresponding to changes in longitude become shorter with increasing latitude. It should be noted that in this form, and in the implemented form, this equation operates in radian units and is valid only for the northern hemisphere.

2.5 Optimization for Uniprocessor Platforms and Time Complexity

Implemented without optimizations, this algorithm design performed poorly in terms of speed (table 2.1). To rectify this performance problem, the implementation was optimized by looking ahead after each rectangle was processed, re-sizing the next rectangle so that its edge is on the boundary of a Stage IV grid cell (unless the boundary of the polygon runs through the same cell). The calculation process here made use of the second FORTRAN-77 function supplied by NCEP to calculate the latitude or longitude of the boundary in question [Lin, 2002]. Table 2.1 shows the comparative times of the optimized code over the non-optimized version: the optimizations produced an execution time performance gain of several orders of magnitude for precisions of 10^{-4} and better.

For both the optimized and non-optimized versions of this algorithm, the computational complexity was itself a complex matter. On polygon vertices, the un-optimized code had to iterate over each vertex once per point check within a subinterval. As the subinterval division was based upon rectangles in two dimensions, the point checking routine was determined to be quadratic on the size of the watershed. Thus, a worst-case trinomial time growth rate was predicted from the theoretical computations. When test runs were executed, however, it was discovered empirically (table 2.1) that a tenfold increase in computational precision (on a constant watershed size) resulted in a 100-fold increase of execution time. Thus, based upon the computational precision alone, the time complexity behavior of the un-optimized algorithm was determined to be quadratic on precision level.

When a similar empirical analysis was performed on the optimized version of the algorithm, it was discovered that increases in precision through 10^{-8} produced only slight increases in the execution time. However, toward higher levels of precision, a tenfold increase in execution time tended to occur with each tenfold precision increase. It was therefore determined that the optimized program has a linear-logarithmic growth behavior ($O(N \log N)$) in terms of precision level, at least for precisions within the limitations of double-precision floating-point numbers. This growth rate amounts to a substantial improvement over the unoptimized quadratic ($O(N^2)$) behavior.

2.6 Testing the Implementation

In a Stage IV analysis, there are 881 rows and 1,121 columns, or 987,601 data points, covering the entire contiguous United States. With no other piece of available software against which to check the algorithm, a qualitative analysis was first made against the graphical version of the Stage IV analysis (figure 2.1), produced daily by NCEP. To ensure that the area of integration was correct, temporary debugging statements were used to calculate the total spatial area of the polygon. For the Reedy-Saluda watershed (shown in figure 2.2), the program returned an area of approximately 2300 km². According to a script run in ESRI ArcMap (ESRI, Redlands, CA USA), the area of the same watershed is 2000 km², based upon a much more sophisticated projection.

Figure 2.1: A graphical Stage IV analysis map, from NCEP

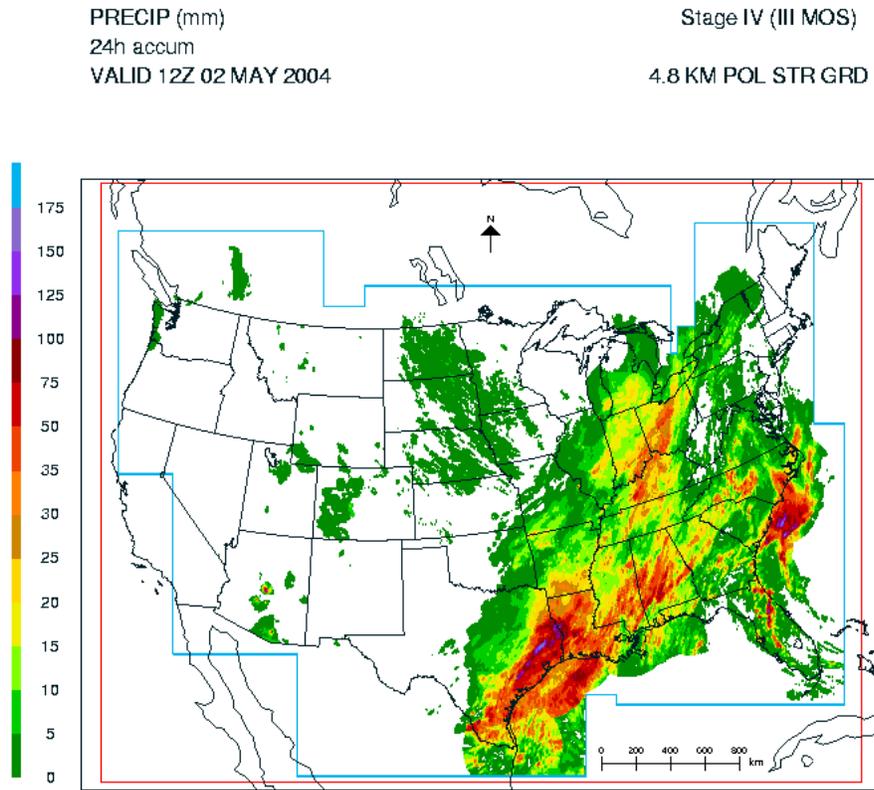


Figure 2.2: A map of the Reedy-Saluda watershed

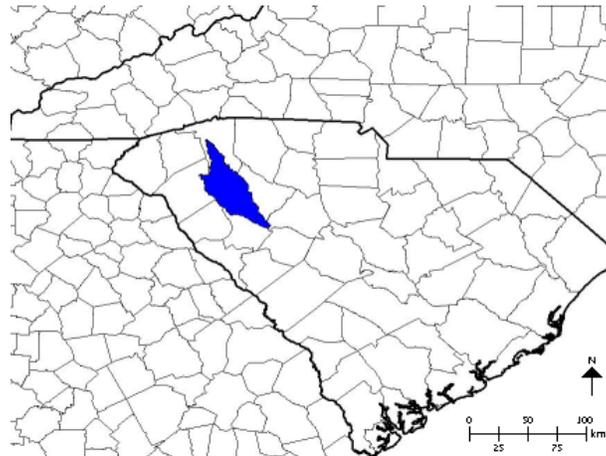
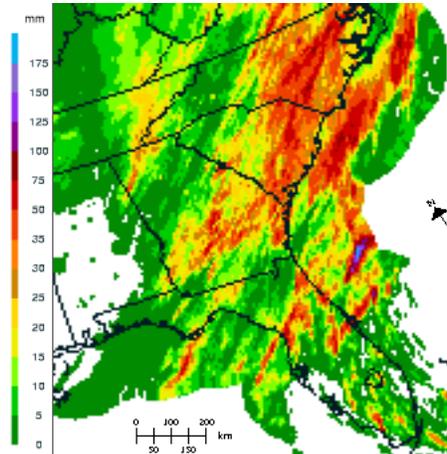


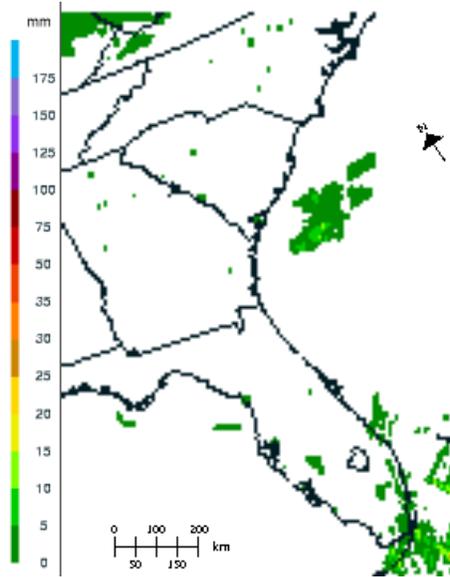
Figure 2.3: Rainfall over the Reedy-Saluda watershed on 3 May 2004



On 3 May 2004, a rainfall event occurred over the Reedy-Saluda watershed near Greenville, SC. The program indicated that approximately 16 billion liters of water fell in the basin, which averages to about 7 mm of rain over the entire watershed. By comparing this average with figure 2.3, it was determined qualitatively that the indicated amount of 16 billion liters was reasonable, because the graphical visualization showed that the watershed was shaded mostly with 0-5 and 5-10 mm accumulations. On 5 May 2004, no rain fell over the watershed (figure 2.4). As predicted, the program returned a value of zero liters.

Performing a numerical analysis on the data proved to be more challenging. Since no other software was available against which to test the developed algorithm, the only available numerical strategy was to compare the program-produced precipitation estimate to the surface observations of ground rain gages. As only a few of these gages were located within the watershed, it was necessary to compare the total averages of gage data and Stage IV estimate data. In order to obtain these averages for the gage data, a simple arithmetic mean was performed. For the Stage IV estimate data, the total amount of precipitation was calculated via the algorithm, and then divided by the basin size (nominally 2300 km²) as calculated during the earlier debugging phase. A qualitative analysis was performed for each of 12 events selected from days with significant rainfall within the watershed. Table 2.2 summarizes the results of the numerical analysis.

Figure 2.4: Rainfall (or lack thereof) over the Reedy-Saluda watershed on 5 May 2004

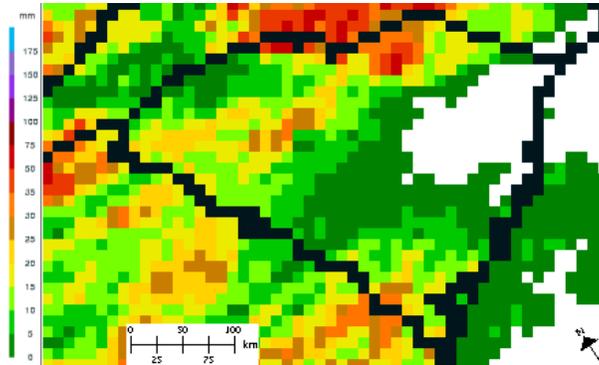


Date	BC1	BC5	KC	LC	OCC	RR20	Avg	Run	RAvg	Diff	Dev
06/24	0.03	0.03	0	0.01	0	0	0.01	27.9	0.47	0.46	3924.30%
06/25	0.09	0.6	0.01	0.03	0.07	0.39	0.2	2.14	0.04	0.16	81.83%
06/26	2.04	2.11	1.6	1.7	1.17	0.99	1.6	51.3	0.86	0.74	46.09%
06/28	0.18	0.9	0.15	0.14	0.11	0.06	0.26	35.6	0.6	0.34	133.11%
06/29	0.01	0.01	0.05	0.04	0.21	0.01	0.06	2.01	0.03	0.02	38.45%
07/01	0.29	0.28	0.21	0.2	1.23	0.32	0.42	21.7	0.37	0.06	13.39%
07/02	0.03	0.02	0.07	0.05	0.88	0	0.18	14.1	0.24	0.06	35.27%
07/03	0.15	0.15	0.06	0.06	0.06	0.04	0.09	35.3	0.59	0.51	584.58%
07/04	0.02	0.03	0.02	0	0.17	0.02	0.04	4.12	0.07	0.03	60.04%
07/05	0.11	0.11	0.01	0.01	0.11	0.02	0.06	7.82	0.13	0.07	113.46%
07/06	0.9	0.89	0.37	0.42	0.95	1.23	0.79	13.1	0.22	0.57	72.31%
07/09	0.79	0.86	0.82	0.85	0.22	0.31	0.64	9.92	0.17	0.47	73.99%

Table 2.2: Comparison of Observed vs. Estimated Precipitation

Key: **Date:** ending date of precipitation estimate/observation; **BC1, BC5, KC, LC, OCC, RR20:** tipping-bucket rain gage measurements from six sites; **Avg:** average of the tipping-bucket gage data; **Run:** estimated total precipitation accumulation (billions of liters); **RAvg:** total estimate (Run) divided by the size of the watershed (hence, average estimated precipitation); **Diff:** difference between the tipping-bucket average and the estimate average; **Dev:** percentage difference between the two averages.

Figure 2.5: Rainfall in South Carolina on 24 June 2004



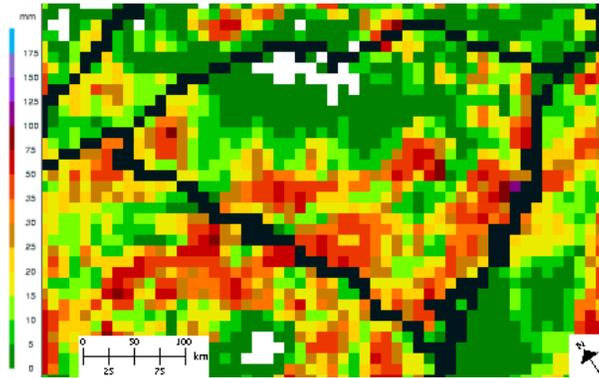
Note that the amount of precipitation that accumulated in the Reedy-Saluda watershed varied because of the nature of the storms that day. Source: NCEP.

Interpreting the results of these 12 events was not easy, as there were variations of several orders of magnitude between the averages. Once again, it was necessary to perform a qualitative analysis on the data. To perform this analysis, the normal summer climatic pattern had to be taken into consideration. In this part of South Carolina, it is typical for scattered summer afternoon convective activity (showers and thunderstorms) to develop in the heat of the day. These pulse-type storms often contain heavy rain, but they normally affect a relatively localized area. It is far less common to have widespread precipitation during the summer in Upstate South Carolina. [Outlaw and Murphy, 2000]

Looking at both the qualitative and quantitative analyses of the precipitation data, it should be noticed that the day with the highest variation in the averages (almost 4000%) also exhibits a pattern of widespread light precipitation (5-10 mm) and a single swath of heavier precipitation (15-25 mm) as shown in figure 2.5. On the day with the least variation in averages (under 14%), a more uniform distribution of rain fell over the watershed, with heavier amounts to the south and west (figure 2.6). Figure 2.7 shows an “average” day, highlighting the pulse-storm tracks as swaths of heavier precipitation. The calculated summation varied from observations by almost 74% on this date.

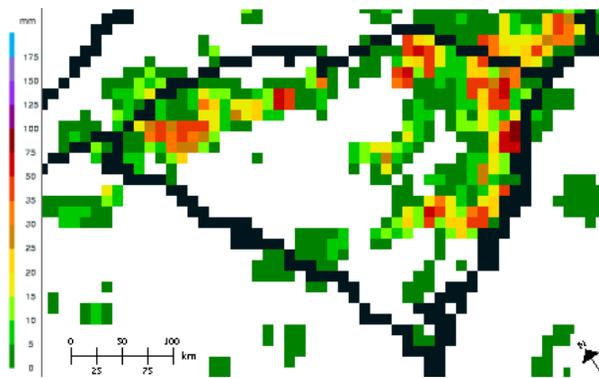
An overall visualization of deviation for the selected events is presented in figure 2.8. It is interesting to note that, while the percentage variations can be quite large,

Figure 2.6: Rainfall over South Carolina on 1 July 2004



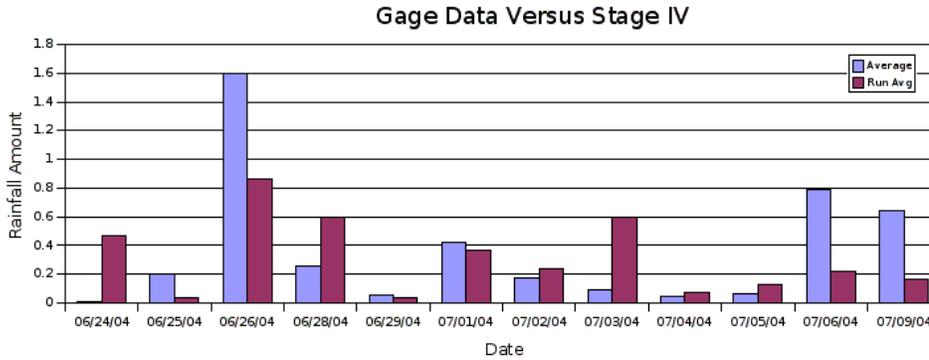
The area of the Reedy-Saluda watershed received a relatively uniform light accumulation that day. Source: NCEP.

Figure 2.7: Rainfall over South Carolina on 9 July 2004



This event was more typical of a summer afternoon storm pattern, where some areas received heavy rainfall along a storm track, while many other locations remained dry. Source: NCEP.

Figure 2.8: Differences Between Gage Data and Estimates



Visualization of differences between recorded gage data and Stage IV estimates. In this presentation, Average is the average of the surface gages, and Run Avg is the average estimated precipitation. Units are inches.

the actual deviations (shown in inches) in accumulation are not as large for events in which less total precipitation fell. This behavior fits well with the expected rainfall pattern for summer-season precipitation in this area of South Carolina. Widespread precipitation tends to be lighter, while afternoon thunderstorms tend to produce heavy rain over localized areas.

2.7 Sources of Error

The primary sources of error in the output of this program appear to be due to the data themselves, to the bounding functions on the polygon, and to the projection used in the differential area calculation. Stage IV precipitation analyses are only single-precision in nature, and while they are quality-checked, they do rely heavily on the Doppler radar for often erroneous precipitation estimates [NOAA, 1990, Ulbrich and Lee, 1999]. At the edges of the polygons, this implementation treats a point lying directly on the edge to be outside the polygon. Exactly how accurately the threshold of interior/exterior points is measured depends upon the precision used, which in turn is related to the running time. Finally, the simplistic projection used for the differential area calculation treats the earth as a sphere and not as an ellipsoid, and the projection used employs a linear adjustment factor.

2.8 Conclusions and Potential Future Changes

The Stage IV total rainfall estimation algorithm provides a single numerical estimate of the total amount of accumulated precipitation within a watershed. With this estimate, it is possible to analyze roughly how much water has entered into the watershed, which in turn provides a means to predict the behavior of the associated river or stream. Such a prediction of water quantity is useful in the prediction of water levels along the stream, as well as predicting water quality. This information, which can be obtained quickly with the optimized algorithm, can be used in research and operational settings to monitor the impacts of urban development, as well as to forecast flooding problems during rainfall events.

In future versions of this software, the differential area calculations could be improved greatly by using ellipsoidal scale factor formulae to produce more accurate values of distance measure for dimensions on the earth's surface. Also, the software could be modified to work with multiple polygons. Further optimizations could be used, especially optimizations that provide some bypasses at values of zero and detect changes in values instead of grid cell edges.

Perhaps the most exciting future work options are in terms of the applications for which this software could be used. One key application area will be in hydrological applications for runoff analysis and water quality modeling. Work already has been undertaken to map WSR-88D radar data and digital terrain data together for this purpose [Reed and Maidment, 1995]; however, the use of combined surface gage and radar estimate data may be of value in future work. Furthermore, this Stage IV data can be used in conjunction with other data, including sensors deployed in the field, to provide a platform for environmental analysis and land use change investigations.

Chapter 3

Soil Moisture Sensor Network

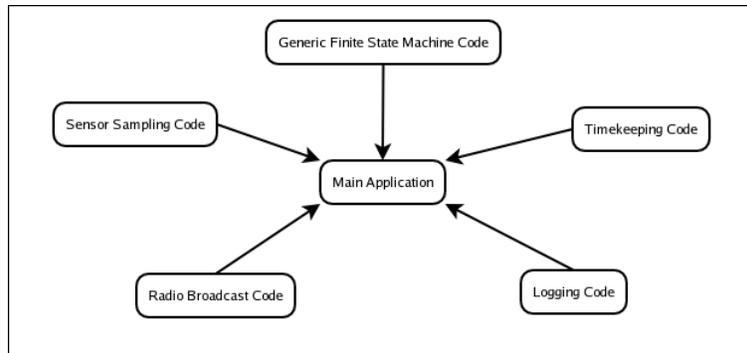
3.1 Overview of the Soil Moisture Network

The primary motivation for this experiment was the detection of ephemeral streams, or channels of runoff that contain water less than 30% of the time [North Carolina Division of Forest Resources, 2004]. Detection of these streams is important for land use analysis and planning, because pollutants that enter ephemeral streams are likely to wind up in major bodies of water. With sensors that can gage the total amount of moisture in the soil, it is possible to identify these streams by finding locations with high moisture content during storm events.

In order to compare soil moisture levels over a geospatial area, an inexpensive means of recording moisture data was needed. In addition, a power-efficient method to take samples and produce these records was required, because grid power was not available at the site. To meet these requirements, Berkeley mote hardware was selected for its low power consumption, its ability to maintain a log of data, and its radio communication abilities for sharing data between nodes [Polastre et al., 2004].

For this experiment, the Crossbow Corporation's Mica2dot mote was chosen. This mote features a low-power micro-controller, a 512 kilobyte flash storage chip, six available 10-bit analog-to-digital (ADC) converter lines, and a 916 MHz radio [Polastre et al., 2004]. Each mote was enclosed in a sealed housing, with its batteries and sensors connected externally. Software was written in the nesC language for the TinyOS operating system, and the software application was installed on each mote.

Figure 3.1: Component Design of the Soil Moisture Application



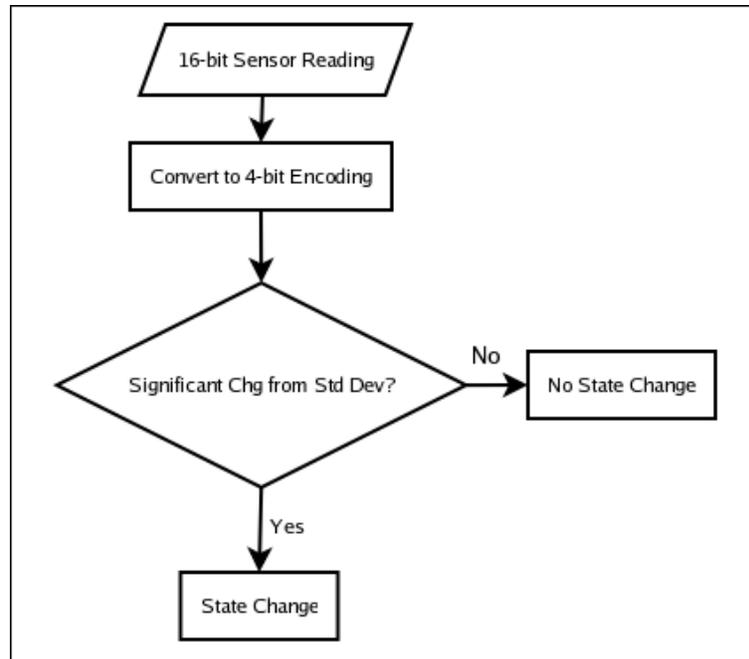
The sensors used in this experiment were Delmhorst model GB-1 gypsum block sensors, typically used with the Delmhorst KS-D1 moisture meter (Delmhorst Instrument Company, Towaco, NJ). Comprised of a probe surrounded by a gypsum buffer, these sensors were designed to reach moisture equilibrium with the surrounding soil. Once equilibrium was reached, the total resistance to current flow was proportional to the total amount of moisture in the soil, and so the mote could measure the voltage drop through the sensor in order to gage relative soil moisture [Delmhorst Instrument Company, 2003].

3.2 The Software Design

3.2.1 Nondeterministic Finite State Machine Model

Underlying the software used in the experiment was the premise that “interesting,” and hence loggable, events only occurred when there was a change in the sensor readings. Whenever the sensors remained at the same level, it was assumed that no event was occurring, and so no logging took place. This assumption was necessary due to the high current draw of writing to the flash chip [Mainwaring et al., 2002]. As a result of acting only upon changes in state, the principal model of software development was based on the concept of a finite state machine. This model has been used in other sensor network designs in the past [Kim and Hong, 2003]. Figure 3.1 shows how the application is designed to take advantage of a general finite state machine implementation.

Figure 3.2: Generic Finite State Machine Decision Process



Because the sensors themselves are imprecise, it was necessary to account for some drift in the outputs. This drift could occur even when the soil moisture did not actually change, so it was necessary for the mote to make a determination as to whether or not any particular change constituted an actual deviation in soil moisture. This determination was made by first encoding the sensor readings on a 0-15 scale. If the average of these encoded readings exceeded the average of the previous set of readings by some number of standard deviations,¹ the transition function dictated a state transition and a logging of the event (figure 3.2). However, when the specified threshold was not exceeded, the function did not produce a transition. As a result of this calculation, the finite state machine model was necessarily nondeterministic, since the sensor inputs alone did not dictate transitions between states.

¹Calculated from the readings of the six sensors at the previous sampling, with $n - 1$ degrees of freedom.

3.2.2 nesC and TinyOS Programming

Due to the relatively steep learning curve for nesC and TinyOS, programming for the experimental network required several months. During this time, it was necessary to create software components that could read the sensors, perform the ADC conversions, log the data, broadcast messages to other nodes, and keep track of the current time and date.

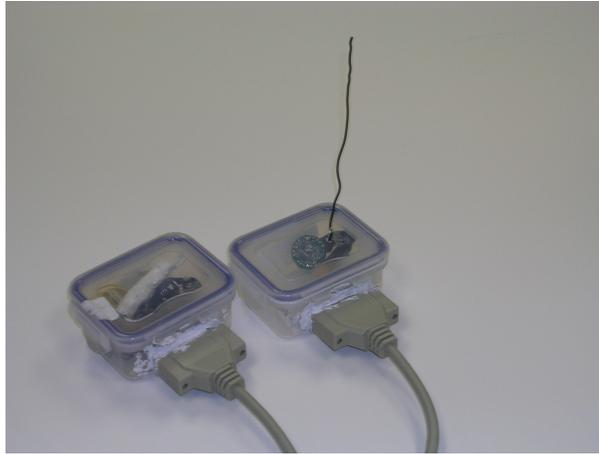
In order to save power, it was necessary to take sensor readings typically at fifteen minute intervals. However, to achieve better temporal resolution, the software was designed to increase sensing frequency to five minute intervals if several nodes were detecting an increase in moisture. In order to accomplish this dynamic change, broadcast messages were employed so that each node could announce when it was getting wet. If such messages were received by any given node while moisture was increasing, then that node would increase its reading frequency until the broadcasts from other nodes ceased. In this way, the network would collectively increase or decrease its sampling frequency as soil moisture increased or did not increase.

3.3 The Experimental Method

Prior to deployment of the network, it was necessary to seal the motes and their external batteries against water intrusion. Because the pins on the Mica2dot motes bent easily, it was also necessary to use a more robust connector for programming, testing, and deployment. As the mote has 18 pins, a 25-pin sub-D connector (DB-25) was chosen to connect the mote to external features such as the programming and power/sensing interfaces. The mote itself was enclosed in an inexpensive waterproof container obtained from a local vendor, with the DB-25 connector mounted on the exterior. Before deploying the network, silica gel desiccant was placed in each mote container, and the container was sealed with silicone.

Power to both the mote and sensors was provided by an external battery pack containing two AA sized lithium batteries. To protect the batteries from the weather, they were enclosed in a semi-sealed container similar to those used for the motes. A DB-25 cable was used to attach a mote, while a 9-pin sub-D (DB-9) connector attached the sensors directly to the battery box. Each sensor was wired to a dedicated pin on the mote's ADC converter and tied to a common ground. At the other end of the sensor wires, each gypsum block was buried under a shallow layer of soil.

Figure 3.3: Mote and Battery Box Setup



All connections were given a coating of silicone sealant to seal out water ingress, and the mote boxes were elevated above ground level to avoid their being placed in flowing water situations. Figure 3.3 shows an example of the mote and battery box setup.

After being prepared, the combined sensor, battery, and mote units were deployed in a section of the Clemson Experimental Forest [Sorrells, 1984]. Fifteen days after the field deployment, the sensors, motes, and batteries were retrieved to the laboratory. The contents of the logs on each mote were downloaded to a workstation and analyzed. A multimeter was used to measure the final voltages of the battery packs in each battery box.

3.4 Results of the Deployment

During the deployment period, which began on 17 February 2005 and ended on 4 March 2005, the surface rainfall observation station in Anderson, SC (KAND) observed measurable precipitation on 20-21 February, 24 February, and 27-28 February [National Weather Service Forecast Office Greenville-Spartanburg, 2005]. This station was the closest site to the deployment area for which data were readily available. Also, because the weather pattern for these days involved widespread regional precipitation, it was assumed that the network observed events on these

Table 3.1: Post-Deployment Data

Mote	BattBox	LogDate	NumEntries	Voltage	Condition
1	6	3/2/2005	10	3.186V	Breached
2	2	(invalid)	0	1.880V	Breached
3	1	(invalid)	0	0.000V	Condensation
5	3	3/2/2005	20	3.283V	Condensation
6	5	3/1/2005	1574	3.164V	Breached

Key: **Mote:** unique mote number assigned to each mote box. **BattBox:** unique battery box number. **LogDate:** date of the last log entry. **NumEntries:** total number of log entries. **Voltage:** final voltage after deployment. **Condition:** conditions inside the battery box.

dates. The results of the deployment are summarized in Table 3.1.

Each battery box contained some water upon retrieval. If the water level was minimal and generally indicated by droplets on the underside of the cover, it was assumed that condensation had occurred. For the three boxes in which measurable standing water was observed in the bottoms, it was determined that rainwater had breached the seals around the connectors. Fortunately, the seals on the mote boxes fared much better, as the insides of each mote housing were completely dry.

Two motes failed to log on the deployment (motes 2 and 3). The failure of mote 3 was obvious and due to human error. One of the batteries was installed backwards, and the mote never received power. Due to the low final voltage of battery box 2, it is likely that the battery box flooded during the first rainfall event, creating a short circuit. It is believed that a hardware wiring failure in a sensor harness is to blame for the behavior of mote 6, with its substantial number of log entries. Further analysis of the data revealed that the majority of entries were zero entries, with the exception of a few nonzero entries believed to be caused by short circuits during one of the rainfall events. Because the standard deviation of zero data is itself zero, the mote logged events at each sensor reading. However, the time stamps of these readings did show that the logging interval did change between five and fifteen minutes during the deployment, confirming operation of the radio communications.

Of the motes that logged properly, none recorded events on each date recorded at KAND. Motes 1 and 5 both recorded events on 20 February, while mote 6 continued

to read all zero. Mote 1 recorded some drying on 22 February, as did mote 5 on 23 February. Mote 5 recorded a trend toward more moisture beginning on 25 February, with data running through 1 March. At about the same time in late February and early March, mote 6 appeared to short-circuit due to the moisture intrusion, as nonzero data was recorded. Some issues with event reading may have been due to problems with the sensors: several of the gypsum blocks partially disintegrated over the fifteen-day deployment, well ahead of expectations.

3.5 Conclusions and Changes for the Next Deployment

Overall, the results of the experimental deployment were both encouraging and discouraging. Many features of the software worked properly, while many aspects of the hardware failed miserably. Water intrusion was a major culprit in the hardware failures. Future deployments will use a single sealed container for both the mote and its battery, minimizing the risk of water intrusion and associated short circuits.

As a result of the observed disintegrations, it was determined that better soil moisture sensors should be used for future deployments. The chosen replacement sensors for the next test will be the Decagon Devices “ECHO” sensor probes, which are accurate to within 3% of soil moisture when left uncalibrated [Decagon Devices, Inc., 2002]. Due to the cost of these sensors, it will be necessary to scale back the number of sensors per mote from six to three.

Attendant with the reduction in the number of sensors, it will be necessary to adjust the transition function to perform a statistical analysis on a time series of data, instead of on the deviations in individual measurements. In addition, some adjustments will be needed to the threshold at which logging occurs. On the initial deployment, 1.0 standard deviations in the encoded data were needed to trigger a transition. However, that number is relatively high and likely caused the Type I error in failing to detect several events.

Once the issues with the existing soil moisture network have been fixed, it will be re-deployed for another test deployment. A future operational version of this network will provide valuable information on soil moisture and ephemeral streams in the deployment area. Moreover, the core software in this application, along with its

nondeterministic finite state machine design, will be reused in future environmental applications. The components that implement this software model will form the building blocks of other sensor networks that monitor changes in environmental parameters, speeding the development and deployment of future sensor network solutions. Future networks based on this design will monitor a number of application areas, including stream turbidity and forest fires.

Chapter 4

Turbidity Sensor Network

4.1 Introduction to Turbidity Sensing

Whenever precipitation occurs in sufficient amounts as to cause runoff, sediment is washed into rivers and streams. The amount of sediment that enters the stream largely depends upon the ground cover in the vicinity. However, once this sediment enters the water, it becomes suspended in solution. This suspended matter can be approximated by measuring the amount of light refracted off particulates, a quantity called “turbidity” that directly affects water quality and clarity [Lewis, 1996]. In Upstate South Carolina, streams with a high level of turbidity often have a noticeable red color, indicative of relatively low water quality and high clay content.

Although spatial variations in turbidity tend to be small when compared to temporal variations [Lewis, 2002], some variation in spatial turbidity still could occur, especially in streams where tributaries add additional sediment. Many streams in Upstate South Carolina contain multiple tributaries, so turbidity measurement over a geospatial area is an important sensor network application. Though it is still under active development, a sensor network for turbidity monitoring is being designed and constructed. This network, which will utilize a number of technologies including motes, pressure transducers, turbidity sensors, and meteor burst communications, eventually will be deployed on the North Saluda River in Greenville County, South Carolina. This river and its associated Poinsett Reservoir provide drinking water for the Greenville Water System [Greenville Water System, 2004]. In this application description, the present and future design issues and objectives will be discussed.

Table 4.1: Total Suspended Sediment (TSS) and Sensor Voltage Output

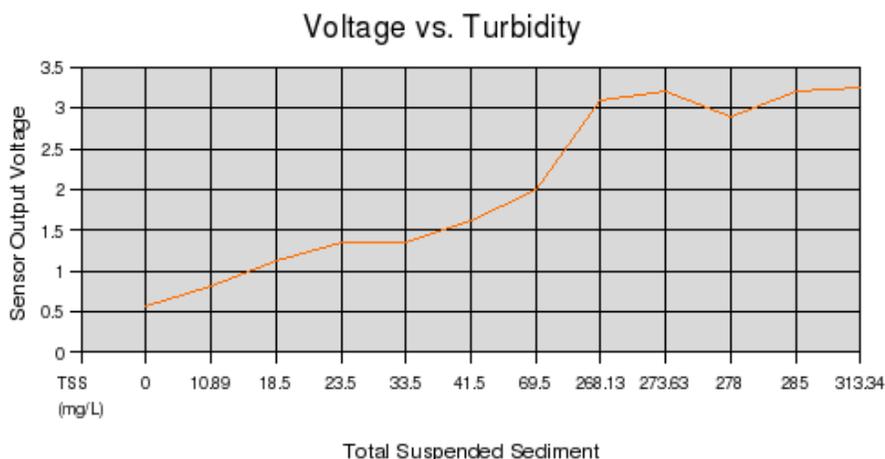
TSS (mg/L)	Voltage (V)
0	0.578
10.89	0.822
18.5	1.118
23.5	1.350
33.5	1.350
41.5	1.630
69.5	2.000
268.13	3.100
273.63	3.200
278	2.900
285	3.200
313.34	3.250

Total Suspended Sediment (TSS) is shown in milligrams per liter of the high-clay soil characteristic of Upstate South Carolina. The voltage output is the measured output from the turbidity sensor.

4.2 Hardware Design

For this application, the MoteIV corporation’s Telos Revision B and Tmote Sky motes will be utilized to provide the sensor network processing power. These motes are well-built and feature Texas Instruments micro-controllers, large-capacity flash storage chips, and on-board USB device connectivity [Polastre et al., 2005]. Honeywell model 50001204 low-NTU laser refracted turbidity sensors, designed for dishwasher applications, will provide the turbidity sensing functions [Honeywell International Inc., 2004]. Two absolute-referenced pressure transducers will be employed. One of these transducers will be located above the surface of the water and will be open to the air, in order to measure barometric pressure. The second pressure transducer will be located above the water line, but a tube will run down into the water and will be open at the end. Changes in the water level will exert pressure on the air in the tube, allowing for the measurement of water depth.

Figure 4.1: Turbidity Sensor Calibration Curve



A preliminary sensor calibration was performed for the turbidity sensor¹, since that sensor was designed for dishwashers and not for stream environments. The calibration process involved the use of a large tank of pump-circulated water, into which red clay sediment solution was added gradually, while voltage output readings from the sensor were recorded. Solution samples were obtained as each voltage reading was taken, and the solution was later filtered. When the filters dried, the amount of sediment suspended in solution was measured by mass, and the Total Suspended Sediment (TSS) was calculated by dividing the sediment mass by the volume of the sample. The recorded voltages were then cross-referenced with the TSS values, in order to obtain a calibration curve. These output voltages were measured on the output side of the sensor, which was powered at 5 volts input in accordance with the Honeywell specifications [Honeywell International Inc., 2004]. A table of TSS and voltage calibration information is included as table 4.1, and a graphical representation of the preliminary calibration curve is included as figure 4.1.

In addition to sensor calibrations, preparations have been made to provide external antennas for the motes. The Telos B and Tmote Sky motes both ship with internal antennas that have effective ranges limited to approximately 20 meters [Polastre et al., 2005]. For this application, 5 DBi gain magnet-mount external an-

¹Calibration performed by C. Smeal, M. Murphy, and G. Taylor

Figure 4.2: PVC Enclosure for Turbidity Sensor Nodes



tennas have been acquired, which will be affixed to the motes via reverse-polarity SMA connectors. It is anticipated that these external antennas will provide sufficient radio communications capabilities for mote deployment spacings of 50 meters or more.

For the turbidity network deployment, lessons learned from the previous soil moisture tests will be employed to ensure that the electronic components remain in environmentally sealed containers. Unit enclosures constructed of schedule 40 PVC pipe will house the mote, all sensors, and the power supplies for both the mote and sensor². The external antenna wire will be routed through a sealed port in the side of the tube. In the stream, the PVC enclosure will be secured to a metal anchor driven into the stream bed, such that the enclosure is vertical with the turbidity sensor itself (on the bottom of the unit) underwater. A photograph of the PVC enclosure is presented as figure 4.2.

4.3 Software Overview

Because the previous soil moisture sensor application has been designed around a generic nondeterministic finite state machine model, much of the software from that application will be reused in the turbidity network. Changes will be made to support

²Enclosures constructed by Dr. W. R. English

the MoteIV hardware, and other minor adjustments will be made to the existing software, in order to utilize the new array of sensors. Two major improvements will be made to the existing software: a provision for distributed logging will be added, and a new networking protocol will be employed for inter-network communications and forwarding of data to an external gateway device.

4.3.1 Distributed Logging

In order to ensure the best data logging and storage safety, a two-tiered logging approach is being developed for recording sensory data. These two tiers consist of a logging segment for storing each event observed by the node, as well as a second segment for recording a composition of events, along with networking statistics, once per hour. This concept is a rough implementation of the principle of lossy data storage [Ganesan et al., 2003]. Each segment of the log is a circular buffer, wherein newer readings will overwrite the oldest readings whenever the segment fills.

To guard against the potential of data loss from a flash failure on a single mote, the hourly data from each mote will be stored to other motes in the network. This distributed backup algorithm will work by sending backup requests to two other motes that are 1 hop away. Each mote will have the capability to store up to three backup messages per hour from other motes, in order to ensure that backups occur without creating excess network congestion if an odd number of nodes are each in a single hop neighborhood. With this algorithm, it will be possible to recover data from the network even if several motes experience catastrophic failures.

4.3.2 Network and Gateway Communications

In addition to providing distributed backup capabilities, the new network software also will have to route messages through the network to a gateway device. This routing must be self-organizing, since the exact layout of the network will not be known until actual deployment [Culler et al., 2001]. A number of different protocols are being evaluated to provide both the 1-hop detection and distributed backup, as well as the routing. At the gateway, a single mote with a known, fixed address will interface the network to a Gumstix Connex single-board computer (Gumstix, Inc., Palo Alto, CA, USA). This Gumstix device will then interface to a meteor burst modem, which will transmit data back to the researchers by bouncing messages off

the ionization trails of the small meteors that continuously vaporize in the earth's atmosphere [Cumberland et al., 2004].

4.4 Deployment Plans

Prior to the operational deployment of the turbidity network at the North Saluda River, several test deployments are planned to debug the software and test hardware. These smaller deployments are likely to occur on the Clemson University campus and at a test site on the Reedy River in Greenville County. Each of these “beta” deployments will be useful in ensuring that the system as a whole functions according to specifications. Additional simulator testing also will be undertaken.

Once deployed at the operational site, the turbidity monitoring network will provide near-real-time data on the amount of suspended sediment in the North Saluda River via meteor-burst communications. As a backup against a failure of the gateway, distributed network backups of hourly logged data will be performed on the site. With these features, the turbidity sensing network will provide valuable water quality information about the North Saluda River. This information, in turn, will be useful for assessing the grade of raw water used to supply the Greater Greenville area. This network is an example of the use of sensor networks to provide important environmental information for environmental assessment and resource planning.

Chapter 5

Conclusions and Future Research Topics

5.1 Research Conclusions

Three major projects have been undertaken in this research experience: an algorithm for estimating total watershed accumulated precipitation, a sensor network for soil moisture monitoring, and a sensor network under development for stream turbidity analysis. Each of these projects attempts to resolve a small, specific issue within the larger fields of hydrology and hydrometeorology. In fulfilling the research needs of these applied disciplines, many computational and technological issues have been resolved in the areas of geospatial radar-derived data processing and sensor network development. In particular, these projects have required the design of software that executes in substantially resource-constrained environments.

Simultaneously, these research projects raise as many new issues as they resolve. In the case of the Stage IV analysis estimate, it is still an unresolved question as to whether a more efficient algorithm exists to produce the precipitation estimate. It is entirely possible that $O(N^2)$ is the best possible theoretical behavior for high levels of precision for this type of problem. Although a suitable optimization to $O(N \log N)$ exists for the existing Stage IV product (derived in large part from WSR-88D radar data), higher spatial resolution may be available in future systems. More advanced Doppler radar technologies are under active development by a number of researchers, and similar algorithms running on data from these newer devices may require different software optimizations or more expensive hardware.

In the area of sensor networks, a primary issue at the present state of the art is that a network must be deployed in order to be tested under realistic conditions. Test deployments, such as the initial soil moisture network deployment, are time-consuming and expensive to effect, especially when the software to operate these networks already exacts a considerable development premium. Trial-and-error is still a major component of the sensor network design process, and it is likely to remain a major component for the next few years.

5.2 Future Research

Several important future research topics are still open in sensor networks and remote sensing. Among these, perhaps the most exciting potential projects are the integration of remote sensing and sensor network systems, and the validation of sensor network systems by simulation. An interesting idea for future research along these lines would be to utilize remote sensing devices and low-bandwidth communications to trigger sensor network operation remotely. This type of setup could be extremely power-efficient for the network in the field, since it could spend more time in a reduced power state whenever the remote sensing device did not detect an event.

The second major research idea, simulator verification of the sensor network, would have a significant impact on sensor network development. Utilizing data from remote sensing devices or similar existing sensor networks, it would be possible to model the effects of actual event data on a network in development. This type of simulation, which is not readily available for TinyOS-based networks (at least not in unclassified simulators), would aid in testing the network prior to deployment. This type of pre-deployment verification would reduce the need for expensive post-deployment changes to the network software. When changes were needed to the software application, this verification model would be able to provide additional testing prior to the upgrade, allowing for a greater degree of confidence when using remote upgrade procedures.

Together, sensor networks and remote sensing devices provide important data about the current state of the environment. By leveraging these two types of systems in concert, it will be possible to increase both the quality and quantity of available environmental data. Furthermore, the use of remote sensing technologies to predict

the behavior and aid in the deployment of sensor networks, will greatly improve the speed at which sensor networks can be developed and tested. With better in-situ sensor networks and better remote sensing devices, a larger amount of environmental data will be available for research and operational analysis. These data will provide scientists with additional clues into the inner workings of the environment, which in turn will lead to better environmental prediction and urban development planning. Additionally, widely deployed sensors will provide advanced detection and warning of hazards such as flooding, chemical spills, and terrorist attacks. These future benefits of advanced environmental assessment technologies have the potential to produce substantial quality-of-life improvements for the human population.

Chapter 6

Epilogue

Support

This research was funded by the Changing Land Use and Environment (CLUE) grant (a congressional appropriation through the U.S. Department of Agriculture), and by the South Carolina Water Resources Research Institute (funded by the State of South Carolina with matching grants from the U.S. Geological Survey). The author's previous investigations into the WSR-88D radar system and its products were partly funded by Shodor Educational Foundation, Inc., through the National Science Foundation's National Computational Science Institute (NSF DUE-0127488).

Acknowledgments

The author would like to thank a number of people who have helped and guided this research along over the past several years. In particular, thanks go to Dr. Chris Post of the Clemson University Department of Forestry and Natural Resources, for his direct involvement as advisor to these projects and his provision of funding for their creation. Thanks also are in order for Dr. Steve Stevenson of the Clemson University Department of Computer Science, whose initial guidance and initial advisory support for the radar research ultimately led to the author's involvement with the hydrological applications.

Additionally, the author would like to thank:

- Dr. Dan Cooke of the University of South Carolina - Upstate (formerly USC-Spartanburg), for his advisory support prior to the author's transfer to Clem-

son

- Dr. Dale Linvill of the Clemson Department of Agricultural and Biological Engineering, for his prior support in guiding the hydrological application focus
- Larry Lee, Bryan McAvoy, Wayne Jones, and Dr. Joe Pelissier of the National Weather Service Forecast Office in Greer, SC, for their assistance with the WSR-88D system analysis and general meteorological issues
- Dr. Robert Geist of the Clemson Department of Computer Science, for his assistance with the polygon boundary detection algorithm
- Dr. Richard Brooks of the Clemson Department of Electrical and Computer Engineering, for his assistance and guidance with the sensor network topology and distributed data design concepts
- Drs. Jason Hallstrom, Jim Martin, Michele Weigle, and Pradip Srimani, all of the Clemson Department of Computer Science, for their guidance on sensor network issues
- Don Lipscomb of the Clemson Department of Forestry and Natural Resources, for his assistance with GIS algorithms and mapping
- Dr. W. R. “Rockie” English, of the Clemson Department of Forestry and Natural Resources, for his assistance in constructing the PVC turbidity sensor enclosures
- Dr. Tom Williams, also of the Clemson Department of Forestry and Natural Resources, for his assistance with the ephemeral stream test deployment and soil moisture detection network deployment designs

Finally, the author would like to thank his colleagues in the Geographic Information Science Research Laboratory, especially those with whom he worked directly on the sensor networks projects: Sucharita Otta, Archana Godavathy, Pinky Tejwani, and Greg Taylor. Also, thanks go to Cherisse Smeal in the Department of Forestry and Natural Resources, for the Total Suspended Sediment calculations on the turbidity sensor calibration.

Bibliography

- David E. Culler, Jason Hill, Philip Buonadonna, Robert Szewczyk, and Alec Woo. A network-centric approach to embedded software for tiny devices. In *EMSOFT*, October 2001.
- B. Craig Cumberland, Joseph S. Valacich, and Leonard M. Jessup. Understanding meteor burst communications technologies. *Communications of the ACM*, 47:89–92, January 2004.
- Decagon Devices, Inc. *ECH2O Dielectric Aquameter*. Pullman, WA, 2002.
- Delmhorst Instrument Company. *Model KS-D1 Operating Instructions*, 2003. URL http://www.delmhorst.com/pdf/ks_d1.pdf.
- Wesley Ebisuzaki. *wgrib home page*. National Centers for Environmental Prediction Climate Prediction Center, Camp Springs, Maryland, 2004. URL <http://www.cpc.ncep.noaa.gov/products/wesley/wgrib.html>.
- Deepak Ganesan, Ben Greenstein, Denis Perelyubskiy, Deborah Estrin, and John Heidemann. An evaluation of multi-resolution storage for sensor networks. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems*, 2003.
- Greenville Water System. Greenville water system. Online, 2004. URL <http://www.greenvillewater.com>.
- Honeywell International Inc. *Installation Instructions for the Low NTU Turbidity Sensor Application Kit*. Freeport, Illinois, 1 edition, 2004.
- Tae-Hyung Kim and Seongsoo Hong. Sensor network management protocol for state-driven execution environment. In *International Conference on Ubiquitous Computing*, pages 197–199, Seoul, Korea, October 2003.
- Jack Lewis. Turbidity-controlled suspended sediment sampling for runoff-event load estimation. *Water Resources Research*, 32(7):2299–2310, 1996.
- Jack Lewis. Estimation of suspended sediment flux in streams using continuous turbidity and flow data coupled with laboratory concentrations. In G. D. Glysson and J. R. Gray, editors, *Turbidity and other sediment surrogates workshop*, Reno, Nevada, 2002.
- Ying Lin. *Q&A about NCEP Stage II/Stage IV*. National Centers for Environmental Prediction Environmental Modeling Center, Camp Springs, Maryland, 2002. URL <http://www.emc.ncep.noaa.gov/mmb/ylin/pcpan1/QandA/>.
- Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless sensor networks for habitat monitoring. In *First ACM Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, September 2002.
- National Weather Service Forecast Office Greenville-Spartanburg. Past weather: Anderson archived data. Internet, Greer, SC, 2005. URL http://www.erh.noaa.gov/gsp/climate/cli_files/anda.htm.

- NOAA. *Federal Meteorological Handbook No. 11, Part B: Doppler Radar Meteorological Observations*. Office of the Federal Coordinator for Meteorological Services and Supporting Research, Washington, DC, 1990.
- NOAA. *Federal Meteorological Handbook No. 11, Part D: Doppler Radar Meteorological Observations*. Office of the Federal Coordinator for Meteorological Services and Supporting Research, Washington, DC, 1992.
- North Carolina Division of Forest Resources. Types of water resources related to forestry operations. Internet, April 2004. URL http://www.dfr.state.nc.us/water_quality/wq_typeswater.htm.
- Douglas P. Outlaw and Michael P. Murphy. A radar-based climatology of july convective initiation in georgia and surrounding area. Eastern Region Technical Attachment 2000-04, National Weather Service, Greer, SC, November 2000.
- Joseph Polastre, Robert Szewczyk, and David Culler. Telos: Enabling ultra-low power wireless research. In *Proceedings of IPSN/SPOTS*, Los Angeles, CA, April 2005.
- Joseph Polastre, Robert Szewczyk, Cory Sharp, and David Culler. The mote revolution: Low power wireless sensor network devices. In *Proceedings of Hot Chips 16: A Symposium on High Performance Chips*, August 2004.
- Seann M. Reed and David R. Maidment. A gis procedure for merging nexrad precipitation data and digital elevation models to determine rainfall-runoff modeling parameters. Online Report 95-3, Center for Research in Water Resources, The University of Texas at Austin, Austin, TX, September 1995. URL http://www.ce.utexas.edu/prof/maidment/gishyd97/library/nexrad/rep95_3.htm.
- Aaron Ricadela. Sensors everywhere. *InformationWeek*, January 24, 2005:2, January 2005.
- Robert T. Sorrells. *The Clemson Experimental Forest: Its First Fifty Years*. Clemson University, Clemson, SC, 1984.
- Carlton W. Ulbrich and Laurence G. Lee. Rainfall measurement error by wsr-88d radars due to variations in z-r law parameters and the radar constant. *Journal of Atmospheric and Oceanic Technology*, 16:1017–1024, August 1999.
- United States Environmental Protection Agency. Environmental protection agency: Water security. Online, Washington, DC, August 2004. URL <http://cfpub.epa.gov/safewater/watersecurity/>.
- Frank Warmerdam. *Shapefile C Library V1.2*, 2003. URL <http://shapelib.maptools.org/>.