

PIPENET: A Wireless Sensor Network for Pipeline Monitoring

Ivan Stoianov
Imperial College, London
ivan.stoianov@imperial.ac.uk

Lama Nachman
Intel Research
lama.nachman@intel.com

Sam Madden
Timur Tokmouline
MIT CSAIL
{smadden, timurt}@mit.edu

ABSTRACT

US water utilities are faced with mounting operational and maintenance costs as a result of aging pipeline infrastructures. Leaks and ruptures in water supply pipelines and blockages and overflow events in sewer collectors cost millions of dollars a year, and monitoring and repairing this underground infrastructure presents a severe challenge. In this paper, we discuss how wireless sensor networks (WSNs) can increase the spatial and temporal resolution of operational data from pipeline infrastructures and thus address the challenge of near real-time monitoring and eventually control. We focus on the use of WSNs for monitoring large diameter bulk-water transmission pipelines. We outline a system, PipeNet, we have been developing for collecting hydraulic and acoustic/vibration data at high sampling rates as well as algorithms for analyzing this data to detect and locate leaks. Challenges include sampling at high data rates, maintaining aggressive duty cycles, and ensuring tightly time-synchronized data collection, all under a strict power budget. We have carried out an extensive field trial with Boston Water and Sewer Commission in order to evaluate some of the critical components of PipeNet. Along with the results of this preliminary trial, we describe the results of extensive laboratory experiments which are used to evaluate our analysis and data processing solutions. Our prototype deployment has led to the development of a reusable, field-reprogrammable software infrastructure for distributed high-rate signal processing in wireless sensor networks, which we also describe.

Categories and Subject Descriptors

C.3 [Special-Purpose and Application-Based Systems]: *real-time and embedded systems, microprocessor/microcomputer applications, signal processing systems.*

General Terms

Algorithms, Measurement, Performance, Design, Reliability.

Keywords: Wireless Sensor Networks, Pipeline Monitoring, Water Supply Systems, Intel Mote Platforms.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IPSN'07, April 25-27, 2007, Cambridge, Massachusetts, U.S.A.
Copyright 2007 ACM 978-1-59593-638-7/07/0004...\$5.00.

1. INTRODUCTION

US water companies are under increasing pressure to improve the management of their ageing assets and optimize operational and capital expenditure. A recent study by the US Environmental Protection Agency (EPA) estimates that water utilities need \$277 billion over the next 20 years (2003-2023) to install, upgrade, and replace infrastructure [1]. Transmission and distribution projects represent the largest component (\$184 billion) of this estimate. The threat of contaminant intrusion due to leaking pipes [2] or malicious human action will further increase the projected expenditure. Repairing and securing this infrastructure requires large investments of money and time, and therefore, it is essential to direct efforts to upgrading critical areas. Unfortunately, identifying the highest priority areas is a non-trivial task, because of the scale and age of the pipeline infrastructure and lack of operational data. Failures of large diameter (12" and greater) bulk-water transmission pipelines are of greatest concern as these are supply critical systems. While such failures are infrequent, when they do occur, they have dire consequences, including loss of life, severe interruptions in service, degraded fire fighting ability, damage to adjacent infrastructure and buildings, and multi-million dollar repair bills.

In this paper, we describe PipeNet, a system based on wireless sensor networks which aims to detect, localize and quantify bursts and leaks and other anomalies in water transmission pipelines such as blockages or malfunctioning control valves. The system is also used for monitoring water quality in transmission and distribution water systems and monitoring the water level in sewer collectors. We are developing PipeNet in stages, as building such a monitoring system is a complex problem with many unknowns. In this paper we report results from the first two stages, during which we evaluated some of the critical components of PipeNet through a real deployment (Stage 1), and developed a series of algorithms for detecting and localizing the exact position of leaks which we tested under laboratory conditions (Stage 2).

Operational pipelines are subject to complex, highly non-linear temporal and spatial processes that make it difficult to differentiate between faults and stochastic system behaviors. This makes detecting failures a challenging task, leading us towards a solution based on integrating remotely captured data from several sources, including: acoustic/vibration signals, velocity (flow) signals, and pressure transient signals. Acoustic/vibration signals are used for detecting small leaks that might be precursors for catastrophic bursts, while the analysis of pressure transients and

velocity (flow) enables prompt detection and localization of larger leaks and malfunctioning equipment such as air valves.

In a previous study [3] monitoring operational transmission mains, we successfully demonstrated that continuously acquired high frequency time synchronized pressure and velocity (flow) data from multiple locations along a pipeline could be used to detect medium to large size leaks and malfunctioning equipment. This proof-of-concept was carried out by storing sensor readings on CompactFlash on a PC-104 embedded PC. Readings were collected manually and analyzed offline.

Thus, PipeNet is the next generation system based on Intel Motes that addresses the limitations of this past work; it includes remote monitoring in near real-time, variable sampling rate and long battery lifetime. PipeNet also offers support for high data rate time synchronized data collection from multiple locations. In contrast, current data acquisition practice within the water industry relies upon portable loggers and a limited number of remote monitoring stations which have low-duty cycle. These remote monitoring stations do not have the capabilities for high-data rate acquisition, local processing or high-bandwidth transmission.

In summary, the key features of PipeNet are:

- A field deployable system of hardware and software that demonstrates the feasibility of measuring and collecting (in near-real time) hydraulic (e.g. pressure), water quality (e.g. pH) and acoustic/vibration data over long time periods. We report on results of a 22 month trial in the city of Boston.
- Algorithms, demonstrated in a pipe apparatus in our lab, that allow PipeNet to cross-correlate and analyze data at a gateway or server level to detect and locate leaks, perform online burst detection, and diagnose operational failures.
- A laboratory tested reusable software infrastructure for deploying signal-processing based applications to facilitate quick prototyping and ultimately allow for in-the-field software upgrades of our pipe monitoring solution.

In the remainder of this paper, we outline the design of PipeNet, the analysis algorithms we developed under laboratory conditions, and the results and evaluation of our initial field deployment.

1. DEPLOYMENT OVERVIEW

PipeNet requires high data rate sampling to accurately detect and localize leaks, aggressive duty cycling to ensure months of longevity, tight time synchronization for accurate data analyses, and long-range communication to get data from pipes in the ground to backend servers in our lab. These specifications required a solution with substantial processing power.

We split the development process into three stages. During the first stage, we concentrated on the development and field validation of a small-scale prototype deployment. This consists of three clusters and each cluster had a single sensing node. The monitoring system was deployed in collaboration with Boston Water and Sewer Commission (BWSC) and it has been continuously operating since December 2004 through extreme environmental conditions. The purpose of the field trial was to evaluate our preliminary design in terms of durability of sensors, sensor nodes and gateways and reliability of communication under real operational conditions and within typical urban

environment. We also obtained valuable information regarding the ease of installation, maintenance requirements and the total cost of ownership. The field deployment consisted of sensors to monitor pressure, pH and water level at three locations. During this initial field testing stage we did not attempt to demonstrate an end-to-end solution for detecting leaks and ruptures, which would require a much larger scale deployment. During the second stage, we concentrated on testing and validating some of the more advanced data acquisition and analysis techniques including time synchronized data collection and acoustic leak detection in a laboratory setting. The results from the field and laboratory tests are currently being integrated into the third stage, which will be a complete real-time monitoring solution. We describe the implementation and results from the first two stages here; we outline the design and requirements of the third stage in Section 5.

1.1 Stage 1: Field Deployment and Validation

The main components of the three tier PipeNet prototype deployed in collaboration with Boston Water and Sewer Commission (BWSC) are shown in Figure 1. The trial aims to evaluate a range of technical issues, including communication, reliability and long-term performance of sensors and packaging, ease of deployment, and cost of installation and maintenance.

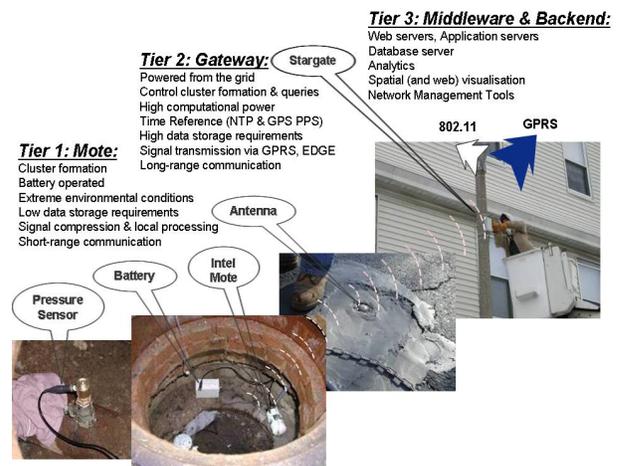


Figure 1: Overview of PipeNet Deployment

The BWSC deployment focused on two major applications: (i) hydraulic and water quality monitoring of transmission and distribution systems including capturing pressure transient events; and pH; and, (ii) monitoring the water level in sewer collectors and combined sewer outflows. Combined sewer systems (CSSs) are designed to collect rainwater runoff, domestic sewage and industrial wastewater. During periods of heavy rainfall, the water volume in a CSS can exceed the capacity of the sewer system. For this reason, CSSs were designed to discharge excess wastewater to nearby streams and rivers. The combined sewer overflows (CSOs), are among the major sources of water quality impairment as the discharge contains human and industrial waste. Nearly 770 large cities in the U.S. (mainly older communities) have CSSs [4]. Three monitoring clusters were deployed, each with slightly different sensors and software configurations:

Cluster 1 includes pressure and pH monitoring sensors in a 12" cast-iron pipe which supplies potable water. Pressure data is collected at 5 minute intervals for a period of 5 s with a sampling

rate of 100 samples per second (S/s). The mote calculates the min, max, average and standard deviation values and sends it to the gateway. pH data is collected at 5 minute intervals for a period of 10 seconds with a sampling rate of 100 S/s. In this case, the mote transmits raw data to the gateway. The data is sent from sensors to a cluster-head via Bluetooth every 5 minutes. For measuring pressure, we modified an OEM piezoresistive silicon sensor that includes an ASIC compensation-based technology that achieves accuracy better than $\pm 0.2\%$, including the effects of non-linearity and hysteresis. The sensor has a start-up time of less than 20 ms, fast dynamic response and power consumption of less than 10 mW. For monitoring pH, we use a glass electrode with an Ag/AgCl reference cell. We developed an immersion apparatus to immerse and position the probe into the centre of the pipe.

Cluster 2 includes a pressure monitor in 8" cast iron pipe. Pressure data is collected at 5 minute intervals for a period of 5s with a sampling rate of 300 S/s. For this cluster, the mote transmits the raw data to the gateway.

Cluster 3 is a water level monitor in a combined sewer outflow collector. As this is a particularly harsh environment, we decided to use hardware redundancy and implement a voting algorithm to identify sensor failures or drifts. For this purpose, we deployed three sensors: two pressure transducers at the bottom of the collector and an ultrasonic sensor on the top (Figure 2).

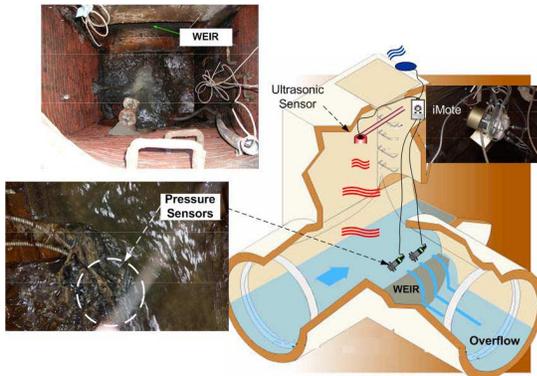


Figure 2: Monitoring the water level in CSOs

The pressure sensors are low-power devices consuming less than 10 mW while the ultrasonic sensor is a high-power device consuming 550 mW. Therefore, we used the pressure sensors for periodic monitoring while the ultrasonic sensor was only used to verify the readings from the pressure sensors when their difference exceeded a threshold or when the water level exceeded the weir height. Data from the pressure sensors is collected in 5 minute intervals for a period of 10 seconds with a sample rate of 100 S/s. Average, minimum, maximum and standard deviation are transmitted to the gateway. We can stream data at a sample rate of up to 600 S/s by remotely reconfiguring sensors (see Section 2.1.)

1.2 Stage 2: Laboratory Validation

A laboratory pipe rig (Figure 3) was constructed to evaluate and illustrate our monitoring solution for detecting and pinpointing leaks using acoustic/vibration data. Though the short length of the pipe does not accurately represent the wave propagation and dissipation mechanisms which occur in large diameter pipelines, it was a convenient way to implement and evaluate our software for pipeline signal processing and time synchronization. The

straight section of the pipeline had a total length of 652cm and diameter of 1¼". The pipe was made of Polyvinyl Chloride (PVC) which has much lower speed of propagation velocity than metal pipes. Leaks were generated under controlled conditions at two locations by installing valves (orifice leaks) along the pipe. Two dual-axis accelerometers (Analog devices ADXL203EB) were attached to collect vibration data generated by the simulated leaks. We experimented with simultaneously connecting these accelerometers to a data acquisition system (HBM Spider8), as well as with connecting each sensor to a Stargate gateway and time synchronizing the gateways through a GPS PPS (Figure 3)

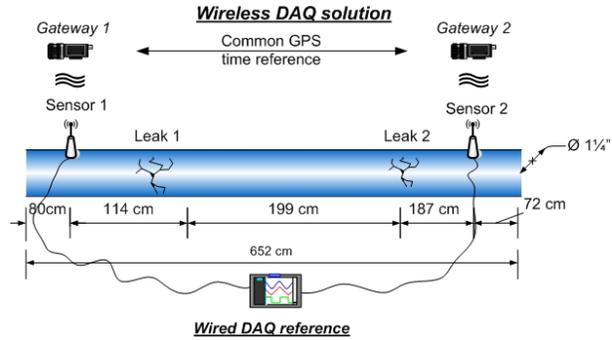


Figure 3: Laboratory pipe rig

2. SYSTEM ARCHITECTURE

In this section, we provide a detailed description of the hardware and software used in the tiered PipeNet system shown in Figure 1. The first tier consists of Intel Motes, equipped with a data acquisition board and a set of sensors. Motes are responsible for the data collection, local signal processing and relaying of data to the second tier via Bluetooth. They are battery operated, so optimal power management and energy conservation are major challenges. The second tier consists of a single board computer (Intel Stargate), which stores and relays data to the backend server (third tier) via a GPRS modem.

2.1 Intel Mote Sensor Node

The Intel Mote platform [5] is an advanced sensor node platform consisting of an ARM7 core, 64kB RAM, 512kB Flash and a Bluetooth radio. We designed a sensor board to interface the Intel Mote to various analog sensors used in PipeNet. The sensor board supports up to 8 analog channels. The analog-to-digital converter (ADC) is connected to a complex programmable logic device (CPLD) which is responsible for driving the ADC clock to achieve the desired sampling rate, and bridging the SPI interface of the ADC to the UART interface supported by the mote.

We implemented Bluetooth scatternet formation and a tree routing algorithms to enable self configuring, self healing networks. In addition, we implemented a lightweight reliable transport protocol to support fragmentation and assembly of large data packets, as PipeNet motes frequently need to transfer messages of up to 100 kB. A detailed description of these algorithms is presented in [5].

The 1st tier is similar in software and hardware architecture to the condition based maintenance deployment mentioned in [25]. We describe the principal differences from this deployment in the rest of this section. These differences fall into two broad classes; First, since we needed to support continuous capture of more data than what would fit in RAM, we were unable to use the store and

forward networking option described in [25]. Instead, we implemented a reliable over the air streaming mechanism. Second, we needed more flexible configuration options from the backend server, storage of configuration parameters in flash, data summarization and local voting features.

Programmable data acquisition configuration: Each mote stores a sampling configuration in non-volatile storage. The backend can relay a new configuration to motes through the Stargate. Configurable parameters include: sampling rate (limited to a max of 600 S/s), sampling duration, sampling channel, and sensor warm-up time. In addition, it is possible to instruct the mote to run the sampling regime with an adjustable period (*periodic mode*), or to run it only once (*burst mode*).

Programmable sleep time: The mote periodically turns on, discovers the gateway, collects the required samples, transmits the data and goes back to sleep for a configurable time period.

Reliable data streaming: Since the available RAM was limited (about 10K left), there was insufficient data storage to continuously buffer samples when using high sampling rates and durations. To support continuous sampling, we implemented a reliable streaming protocol based on the reliable transport protocol mentioned in [5], but with multiple buffers to compensate for possible throughput and latency variations. This enabled us to continuously stream data at rates of up to 600 S/s.

Data processing: For our field deployment, we added the ability to calculate and transmit a summary of a large sample that consists of the min/max/average values for pressure and PH data.

Triggers: The mote can be configured to trigger data acquisition on a channel when a monitored channel exceeds some threshold. This feature is needed so that lower energy sensors activate higher energy sensors when needed. This feature is used to trigger the ultrasonic sensor, providing sensor redundancy when the two pressure sensors disagree or their values exceed a threshold. The triggers and threshold values are remotely programmable.

2.2 Stargate Platform

The motes communicate from the manhole to a Stargate [6] based gateway deployed on a nearby lamppost (Figure 1). The Stargate, the GPRS modem and 802.11 radio are powered from the power lines at the lamp post. An Intel mote is connected to the Stargate through a UART interface; it acts as a bridge between the Stargate and the motes on the pipes. This *cluster head* is responsible for forming the sensor network, converting the configuration data coming from the Stargate and passing it to the correct sensor node, and for delivering data collected via the reliable transport protocol to the Stargate where it can be converted back into data files. These files are periodically sent to the backend via the GPRS link. The Stargate is equipped with an 802.11 link to facilitate drive-by access for on-site configuration and debugging.

2.3 Backend Server

Data is relayed via the GPRS modem on the Stargates to a backend server running in our lab. Data transfer is handled via standard Linux tools: a cron script periodically initiates a connection from each Stargate to the server, and the secure copy tool (scp) is used to transfer data files containing sensor readings. After an scp transfer has completed successfully, the files are deleted from the Stargate. This ensures reliable delivery, as scp computes an end-to-end checksum over the file on the sender and receiver and ensures that they match before signaling that a file

was successfully transferred. The data files are then loaded in a Postgres database that stores the individual sensor readings. Users can browse these sensor readings by connecting to an Apache Web server running on the server. The web site uses Google Maps/Google Earth to allow users to select and browse the sensor locations of interest. Once users have selected a sensor location, they can retrieve data corresponding to a user-specified date / time range and sensor type and can visualize the data [7].

2.4 Reusable Dataflow Processor

PipeNet required constructing several high data rate, signal processing-based applications. To facilitate this construction, we developed a software toolkit, the *PipeNet Analysis Engine*, that makes it possible to rapidly build applications as data flow graphs of signal processing operations [8]. These data flow graphs consist of a collection of “boxes” or “operators”, each of which represents a data acquisition, analysis, or transformation step in the processing of PipeNet data. Data samples are passed between operators as n -ary tuples, where each field of a tuple is a primitive data type such as an integer, float, string, or array. Operators may add additional fields to tuples, filter out tuples, create new tuples, or modify fields of tuples. Figure 4 shows an example of a data flow graph that uses a power-spectrum decomposition (PSD) to detect the presence of a leak from an accelerometer. Boxes represent operators, and arcs the flow of data between operators; arcs are labeled with the types of the tuples that flow along them.

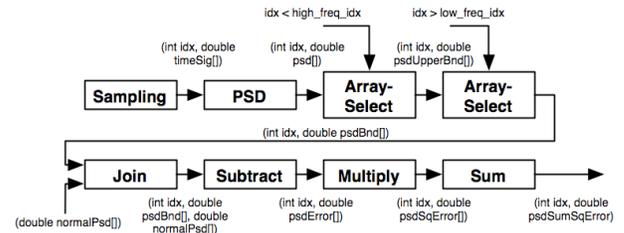


Figure 4: Dataflow representing local leak detection algorithm

Application developers can then build data flow graphs using XML and pass them to the PipeNet Analysis Engine for execution. The Analysis Engine runs in Linux, and hence can be placed either on the backend server or on the Stargates. It includes a wide range of operators, including FFT, Wavelet, and power-spectrum decompositions, relational join and select, a number of signal processing filters, and a range of arithmetic operators. Due to space constraints, we omit the data flow graphs for all of the analyses described below; see [8] for details. We are extending this data flow system into a fully featured stream data processor with a novel query interface called WaveScope [9].

3. DATA ANALYSIS OVERVIEW

This section describes the analytical steps we have developed for processing the data. These analyses ran in our lab deployment but were not deployed in the field. Figure 5 presents a high level description of the different analysis paths. The first path is for detecting large bursts and leaks via pressure and velocity (flow) sensors. The second path is for detecting smaller leaks via acoustic/vibration data. The third path concentrates on a different set of applications such as monitoring water quality in transmission and distribution water systems, and monitoring the water level in sewer collectors.

The first analysis path transmits readings at high rate to gateways for processing and near-real time detection of bursts. Because of the power consumption demands of the continuous sampling and frequent transmission, these nodes need to be powered and hence installed at locations such as pumping stations or reservoirs where power is available or at locations where solar charging systems can be used. A relatively small number of such sensors are needed as large ruptures generate pressure pulses that can be detected over a distance of several kilometers.

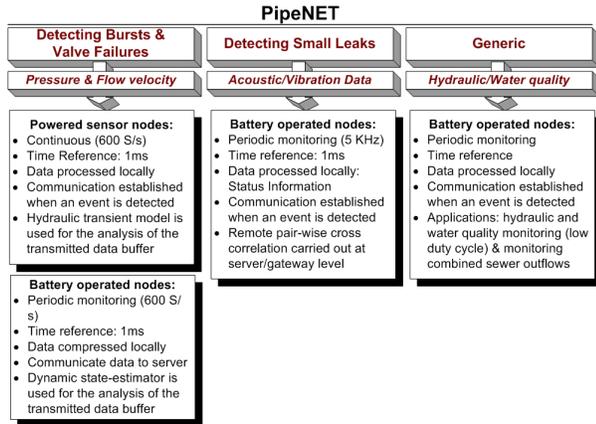


Figure 5: Monitoring system for water transmission pipelines

The second analytical scenario includes the periodic burst acquisition of pressure transients generated by operational variations such as changing the pumping regime which frequently occur (e.g. several times a day). These events are generally scheduled and this allows sufficient time for remotely defining the sampling regime for the battery operated sensors. The acquired data (600 S/s over a period of 5 minutes depending on the length of the pipeline) are compressed locally and sent to the server for centralized data analysis using a transient state estimator [10].

In the second analysis path (Figure 5), smaller leaks are detected by analyzing acoustic/vibration data acquired from densely spaced hydrophones (separated by about 600 m, SR 5KHz). As the analysis for small leaks is not time critical, it is carried out during hours of low background noise (e.g. 2-4 am) for short periods of time (3-5 minutes).

A third type of analysis is used to detect blockages and overflow events in combined sewer collectors. We describe these first two analysis paths in the following subsections. We summarize the results from all three types of analyses in Section 4 below.

3.1 Pressure Data Analysis

A detailed description of the first analysis path is outlined in Figure 5. Pressure signals are continuously acquired at a sampling rate of up to 600 S/s depending on the required accuracy. In our current implementation, the mote continuously streams data to the gateway. Figure 6a shows a controlled burst event that was captured during a preliminary trial on a 36" cast iron pipe. Such bursts (large leaks) generate a low pressure pulse which propagates along the pipeline, as shown by the substantial pressure drop around $t=320$ in signal 1 in Figure 6b. On the gateway, the acquired data is processed using a simple integer Haar wavelet transform to detect pressure pulses (Figure 6c). The integer Haar wavelet transform uses low and high frequency

decomposition filters and does not require floating-point operations. In future implementations, data will be processed on the powered motes without being stored. When the wavelet coefficients exceed a threshold, the most recent data samples will be written to flash and sent directly to the server. On the central server, further analysis can be done to model pressure wave propagation using data from multiple sensors. This makes it possible to confirm the occurrence of a burst and approximately detect its location [10][11][12]. The analysis can also be used to detect malfunctioning air and control valves which can cause deterioration in the pipeline infrastructure.

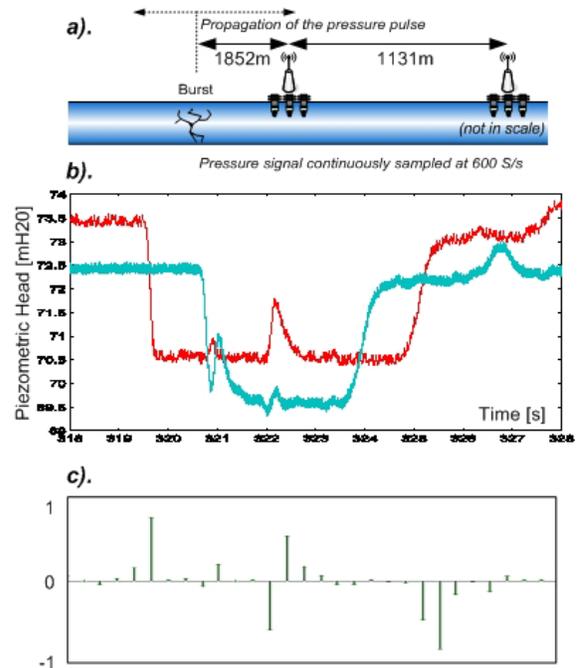


Figure 6: Haar wavelet coefficients (signal 1) used to detect and trigger acquisition and communication of a pressure pulse

3.2 Acoustic/Vibration Data Analysis

The second analysis path is based on analyzing acoustic/vibration data to detect and locate small leaks which are difficult to identify using hydraulic data. We adapted existing algorithms for performing leak detection and localization via cross correlation of acoustic/vibration signals [13][14] to be energy efficient by not requiring continuous transmission of the sensor signals from all nodes to a centralized location and implemented remote time synchronization. For our lab-based experimental setup we acquired vibration signals from accelerometers installed along the pipeline. The same type of analysis with appropriate analytical modifications to reflect differences in the mode of wave propagation and velocity will be used in future real-world deployments. The main advantage of our approach compared to current practice is that the system could be deployed as a permanent monitoring solution which can carry out remote analysis and cross-correlation.

Leaks typically manifest themselves in the acoustic signal as relatively high-magnitude noise in frequency bands that are characteristic of the type and placement of the pipe [15]. These characteristic noises, which propagate uniformly in both

directions away from the leak, are generated by escaping water flowing through the rupture in the pipeline. Generally speaking, if there is no leak or other source of audio signal in the pipe, the signal at different sensors will be uncorrelated. However, if there is a leak, the sensors should all “hear” it, although the signal received by the different sensors will be slightly offset in time depending on their locations relative to the leak. Many of the methods proposed for time delay estimation t_{delay} , between noisy signals which are received at two or more spatially separated sensors are related through a generalized cross correlation approach [14][15][16]. We used a modified technique which has been proposed for time delay estimation when both the source and noise spectra are uncertain [17][18]. The time delay between the two signals, the distance between the sensors and the speed of propagation of sound in the pipe will allow us to compute the location of the leak relative to the sensors.

Unfortunately, to continuously apply cross correlation in a wireless sensor network, we would need to stream data from all sensors to a common node/back-end server where the correlation is applied. Since this would consume large amounts of energy, we would rather exchange signals between nodes only where there is a strong indication that a leak may be occurring. Hence, we desire a technique whereby a sensor can locally determine that a leak is likely occurring with minimum additional communication. To perform this local analysis, the sensor nodes look for increased energy in the frequency bands that are characteristic of leaks. When such energy is detected on a single sensor node, that sensor can compress and send a short duration signal to a central server or a near-by sensor, which can perform the cross correlation operations described above. More precisely, the combined algorithm for leak detection and localization works as follows:

- Nodes monitor their local leak status signal at pre-defined sampling intervals during hours of low background noise (e.g. 2,3,4 am), using an FFT to measure the energy (power spectrum) in the frequency bands that are known to be characteristic of a leak. In our lab deployment, we experimentally determined that these are the 70-140 and 200-250 Hz bands, as shown in Figure 7. This local analysis is shown in more detail in Figure 4.
- If the difference between the energy in these bands in the current signal and the energy in a previously recorded leak-free signal exceeds a preset threshold T , the node sends an alarm to the gateway, which is forwarded to the backend.
- When the backend receives the alarm, it sends a request to the gateway controlling the mote that generated the alarm and its neighbors to collect 60 seconds of data collection. The collection is requested to start at a specific time to ensure that the samples from each of the motes are synchronized. A high accuracy time synchronization algorithm is run in the SN to ensure sub-millisecond synchronization, before the sampling is started.
- When the sensor is done collecting the data, it compresses the data and sends to the backend for cross correlation with adjacent sensors.
- Whenever the backend receives the data from all the addressed nodes, it applies cross correlation on the readings generated by the motes.
- The backend determines the location of the maximum peak in the cross correlation signal. If this peak exceeds a

prescribed threshold P , it uses the time-lag of the peak to compute the location of the leak, as follows:

$$loc_{\text{leak}} = (loc_n + (loc_{\text{neigh}} - loc_n)/2) \cdot t_{\text{lag}} \cdot v_{\text{soundPipe}}$$

Where loc_n is the position of sensor n , in meters, along the axis of the pipe, loc_{neigh} is the position of the neighbor node with which cross correlation was performed, t_{lag} is the time lag of the peak in the cross correlation signal, and $v_{\text{soundPipe}}$ is the velocity of the speed of sound in the pipe material. This equation assumes that node n is closer to the “zero” location on the pipe, and that the time lag in the cross correlation is smaller as the signal gets closer to node n .

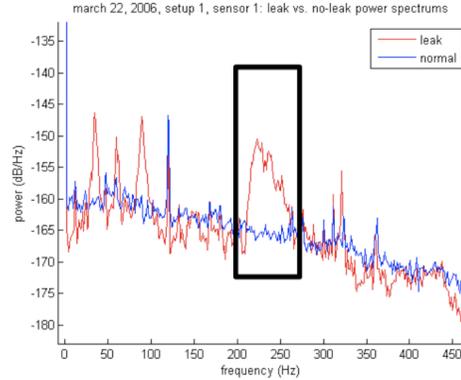


Figure 7: A leak manifests itself as additional energy in certain frequency band

We determined the values of the thresholds T and P experimentally, by analyzing a number of known-leak signals and choosing the peak height such that about 95% of the leaks would be detected according to both thresholds. In practice, the data acquired over a period of time will be used to automatically derive threshold levels based on the noise characteristics of the specific location and signatures from detected leaks.

Determining $v_{\text{soundPipe}}$ is non-trivial as waves propagate through pipes in a variety of different modes [20][23] and it also depends on the air content and temperature. We currently derive $v_{\text{soundPipe}}$ from the velocity of propagation of the pressure transients. We anticipate that monitoring specific pipelines over a period of time will allow us to build numerical models which accurately predict $v_{\text{soundPipe}}$ and the mode of propagation. For our preliminary work, we assume, as has been done in previous work [13][14][20], that the primary mode of propagation is the fundamental mode, where the leak noise is transmitted as a longitudinal traveling wave along the central axis of the pipe. In this case, the speed of propagation depends on the amount and type of fluid in the pipe, the bulk modulus of the pipe material, and the pipe diameter. We have experimentally measured this to be about 1400 m/sec in our pipe. This number is substantially higher than the propagation speed in similar pipe materials reported by Hunaidi et al.[13] and Gao et al.[20] who determined the propagation speed to be about 480 m/sec in their 200m plastic pipe. Although our pipe is also plastic, it is more rigid pipe because of the shorter length (6.5 m) smaller diameter and thick wall. All of these factors are expected to substantially increase the wave propagation speed. We have also observed substantial variations in wave speed in our pipeline; we hypothesize that this is due to the support configuration, as Hunaidi et al. [14] report more uniform propagation speeds over much longer lengths of pipe embedded in the ground.

4. RESULTS

In this section, we discuss the lab and field deployment results.

4.1 Lab Results

To measure the performance of our leak detection and localization algorithms, we simulated leaks at two locations along the pipeline, as shown in Figure 3. In this deployment, the leak locations were fixed, but we could vary the locations of the sensors to measure the effectiveness of our algorithms with different sensor positions. For these experiments, we took traces of data collected from the two sensors, and measured the effectiveness of our local leak detection and leak localization algorithms. For our preliminary assessment of the algorithms, we setup two monitoring clusters and streamed raw data (600 S/s) from the Intel Motes to the Stargates and then transferred the data to a PC where we carried out the analysis. We also used the Spider8 DAQ as a benchmark for the same tests as we acquired data with a sampling rate of 4.8 kHz. For each of 3 leak settings (no leak, leak valve one open, and leak valve two open), we varied the distance between the sensors in six increments between 1.25 meters and 3.4 meters. We ran ten trials with each sensor placement and each leak setting. Each trial consisted of a recording of 30 seconds.

4.1.1 Leak Detection

To measure the effectiveness of our leak detection algorithm, we took all of our data traces and divided them into non-overlapping segments. We then labeled each segment as “leak” or “no-leak”. We selected one of the no-leak segments as a base segment that other segments would be compared against, and took the difference between the power spectra in the 70-140 and 170-240 Hz bands of all the remaining segments and this base segment. To compute the difference threshold in these frequency bands that would best separate the leak and no-leak cases, we separated the segments into equal sized test and training sets. We then used a decision tree classifier to find the best linear separators between the leak and no-leak training data, and used this classifier to predict the leak and no-leak values of the test data.

Figure 8 shows the training data plotted according to the sum of squares differences between the energy in each segment and the base segment in the frequency bands that we identified as being relevant to leaks. Here, the no-leak and leak points are indicated as zeros and plusses, respectively. Notice that leak points have substantially more energy in these two bands.

We learned a decision tree classifier with linear separators that attempts to maximize the number of correctly classified points in the training data set. We then applied this classifier to the test data. We repeated this process of separating test and training data randomly, learning a classifier on the training data, and testing 100 times, and found that the classifier was able to correctly classify 87% of the points across all test data sets ($\sigma = 1.74\%$). As Figure 8 illustrates, the misclassifications tend to occur when detecting small magnitude leaks with small peaks. Such leaks are less important to detect and repair immediately. Furthermore, by setting the classifier thresholds appropriately (e.g., in the region where the x_1 sum-squared differences are > 2500 and x_2 sum squared differences are > 4000), it is possible to create a classifier that has more false negatives but has no false positive detections.

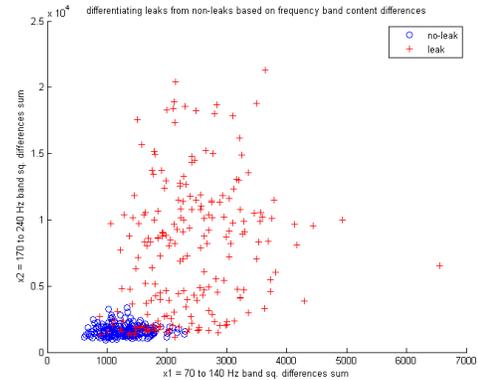


Figure 8: Differentiating Leaks based on differences in frequency band content (sensor 2)

4.1.2 Leak Localization

To measure the effectiveness of our algorithms, we measured the ability of our cross correlation algorithm to successfully localize the leak in a pipeline when two sensors detect a leak. We use the same data as in the leak detection experiments, except that we only cross correlate leak segments. In all cases, these segments were collected at the same time from the two sensors. Because the speed of propagation in our pipes is around 1400 m/sec, and we are sampling at 600 S/s using the Intel Mote, the best accuracy we expect to get is the distance traveled in one sample period, or about 2.3 m. This theoretical accuracy is reduced to 0.3m when using the data from Spider8 DAQ acquired at 4.8 kHz. Figure 9 shows the mean and standard deviation of the error at different sensor positions across all leaky segments when localizing leak 2. Note that the error is about .2 m on average, with the standard deviation ranging .1 and .8 m when using Spider8. The standard deviation is particularly high where the sensors are spaced far apart as the speed of propagation at this separation appears to vary depending on the trial. This could be due to the proximity of the sensors to the pipe ends and the effect of standing waves and their reflections. Results for leak 1 are similar.

4.1.3 Discussion

The above results show that both our local leak detection algorithms and our leak localization algorithms are fairly effective. Though local leak detection is not perfect, it detects a large fraction of the simulated leaks in our experimental pipeline. Our leak localization algorithm, though not error-free, localized leaks to within 30 cm when sensors were separated by 3m or less. Unfortunately, it is likely that the short length of our experimental pipe causes the traveling waves to superpose and form standing waves which can cause the observed variations in the wave speed. However, we are optimistic that the software and hardware we have developed will eventually provide a novel, low-cost solution for acquiring long-term data from field sites which will help to understand the physics of wave propagation and lead to the development of more efficient signal processing techniques. Long-term monitoring of pressure and acoustic signals from a particular pipe will also facilitate the development of more accurate pattern recognition and classification models.

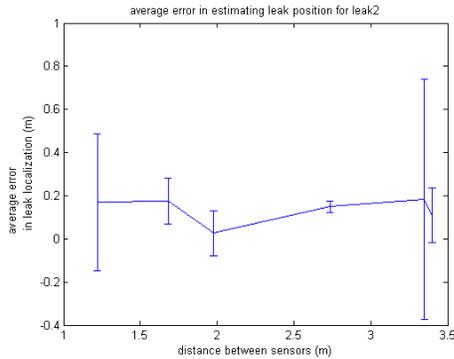


Figure 9: Localization error versus sensor separation

4.2 Field Results

The field deployment of our PipeNet prototype in collaboration with BWSC provided invaluable insight into a range of issues such as communication (Bluetooth & GPRS), ability to capture transient pressure waves, long-term system and sensor performance, and power management and reliability. During the initial stage (December, 2004 – July, 2005) we observed a series of problems with the gateways ranging from unexpected GPRS modem power modes to corruption of the Linux kernel. Detailed analysis of these problems identified design faults with voltage regulators and the watchdog timer on the Stargate platform. An external watchdog and automated reset feature were added to the gateway nodes to monitor gateway performance. The external watchdog timer reboots the gateway once every 24 hours or when the application software halts. Adding these features eliminated the observed problems and reduced the risk of unforeseen problems in the gateway software that required manual intervention by an operator. The gateways were replaced in July 2005 and have been operating since (nearly fifteen months of continuous, trouble-free operation.) The Intel motes have been operating successfully without hardware failures for nearly two years. As mentioned earlier, a reliable transport protocol is used to transfer the samples from the motes to the gateway to ensure the receipt of all the samples from a collection cycle. However, since the motes couple data acquisition with communication, if the mote is unable to connect to the gateway, it skips the collection cycle, goes back to sleep for the configured interval and tries again upon wakeup. However, all stored data on the gateway is eventually transmitted over GPRS to the backend server, as the gateway archives the data if a connection to the server cannot be established.

Data transfer reliability was one of the main parameters that we wanted to investigate in the field deployment. Figure 10 shows the packets received over the time span of the deployment (about 500 days). There are clearly long time periods where we had no data received. This was mainly due to the fact that we were not able to change the batteries for long periods of time (e.g. 5/14/06 – 8/30/06). A vertical line of missing data can be seen right after midnight. This is due to rebooting the gateway at mid-night every day using the watchdog timer to ensure that we recover on daily basis from any potential lockups. However, during the reboot cycle, the motes couldn't communicate with the gateway and the data got dropped in the first collection cycle after the reboot.

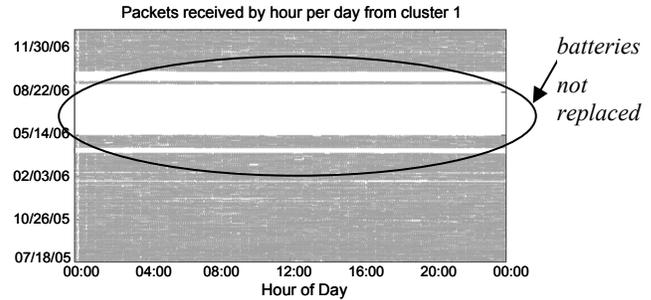


Figure 10: Data transfer reliability in cluster 1

We were concerned with the effect of weather on data communication, so we used data from the National Weather Service to identify weather events and compared it with packet loss data from the operational periods (days with some data transmission, i.e. batteries were not dead). Figure 11 shows the CDF of loss rate per day from cluster 1. The bottom curve shows the packet loss during snow days (>1 inch of snow), which clearly demonstrates that snow has an effect on packet loss (since there were only eight such snowy days during our trial, we view this effect as preliminary.) This increased loss is most likely due to the road-mounted antennas being buried under snow. The second line from the bottom shows the packet loss rate during rainy days (> .3 inches of rain). This effect is less severe than in the case of snow.

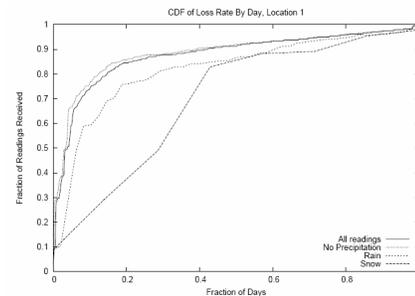


Figure 11: CDF of loss rate for cluster 1

We also calculated overall statistics for the three clusters which we present in Table 1. Since we configured the sleep time to be 5 minutes, the inter-arrival time between collections is about 6 minutes due to network formation (BT inquiry), data transfer and reboot delays. This results in a maximum of 240 collections per day. 87% of the samples in cluster 1 were successfully received if we consider the operational days (days with functional batteries). This is much higher than the 62% observed in cluster 2. We believe this is mainly due to antenna problems. After installing cluster 2, the road was resurfaced which destroyed the embedded antenna resulting in a larger number of non-operational days. After this incident, we installed an antenna under the cast-iron manhole cover which has a thickness of 4.5cm. This change had a poor effect on the communication reliability as the overall percentage of received connections dropped down to 42%. The performance significantly improved after we re-installed the antenna in the road surface. The performance of cluster 3 (CSO) was similar to the performance of cluster 1 despite the particularly harsh environment of operation including occasional chamber floods. The performance of cluster 3 drastically deteriorated after a period of 6 months. The reason for this sudden change was traced to a failure in the antenna connector following an

inspection visit by field maintenance staff. We attempted to repair the antenna connector in field but the repair could not secure the water tightness of the initial installation. This is the reason behind the smaller number of operational days compared to cluster 1.

Table 1: Cumulative Data Loss Rates

Cluster	1	2	3
Number of days	553	524	553
Number of operational days	399	267	269
Total collections received	83238	39548	46708
Total data received (kB)	166476	118644	93416
Percentage of received collections during operational days	87%	62%	72.3%
Percentage of received collections	63%	31%	35.2%

The battery life (6V 12Ah battery) has been consistent with duration of around 50-62 days. The Intel mote consumes 2 mA in sleep mode; 16 mA when sampling from the pressure sensor and A/D board; and about 30 mA when sampling and transmitting. The relatively short battery life is due to the aggressive data acquisition and communication cycles we have chosen (note that the Golden Gate Bridge deployment, which was a high rate sensing deployment based on the Crossbow Mote hardware, lasted only 23 days on an 18Ah battery [21]) We expect that separating the acquisition from communication and using communication intervals of 15 mins with adaptive data acquisition and storage will increase the battery life to beyond one year. The performance of the pressure sensors exceeded our expectations. The sensors have been operating since December 2004 under extreme environmental conditions. The pH sensor, however, has required maintenance or replacement, with a periodicity of two weeks to six months.

Data from the pH sensor over a period of one week is presented in Figure 12. BWSC raises the pH and alkalinity of the water to reduce its corrosiveness. This is deliberately done in order to minimize the leaching of lead and copper from service lines and home plumbing systems into the water. BWSC maintains the pH within 8.6-9.4. The data from our in-line sensor demonstrates the daily cycle of the variations in the pH. The pH is high during hours of high consumption (high flow velocity and low residence time of the treated water in the pipeline system) and low during hours of low consumption and longer residence time. BWSC collects and analyzes pH samples from 26 community locations on a biweekly schedule to measure pH levels. The system we demonstrated could successfully provide near-real time periodic data which is invaluable for the optimal control of the level of the pH and the dosing of the sodium carbonate (Na_2CO_3).

The monitoring system was also successful in accurately capturing several critical operational events such as the emergency failure of the power supply for the Deer Island Sewage Treatment Plant in Boston on the 15th of October (Figure 13) when approximately 25 million gallons of untreated sewage were released into Quincy Bay [22]. In this particular case, the availability of a larger number of deployed sensor nodes would have provided near real time information that would have allowed water utilities to utilize spare capacity of under-full collectors to significantly reduce the discharge volume during this event.

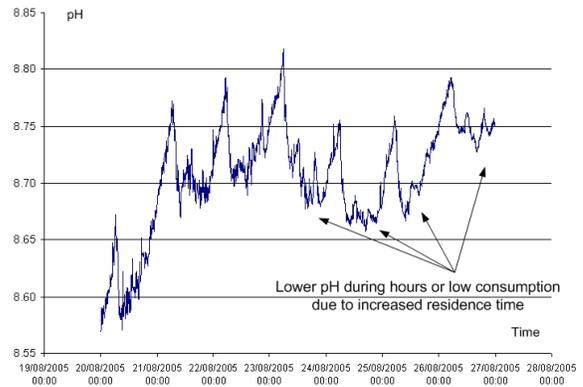


Figure 12: pH Data

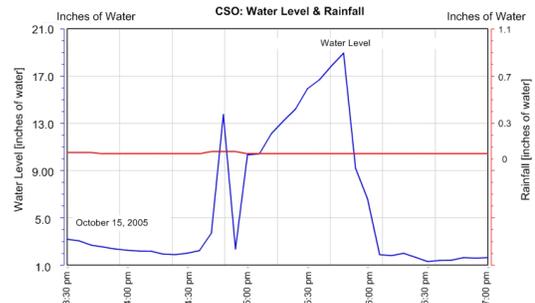


Figure 13: Rupture event detected in Boston

4.2.1 Summary of Lessons Learned

The field deployment helped us to identify a number of important lessons that are of critical importance as we move towards a large scale deployment, including: **Resilience to harsh environmental conditions:** The sensor nodes operated well under extreme weather and installation conditions in Boston. **Separating data collection from communication:** Due to the large data samples and the lack of sufficient memory to store the data and forward it later, we coupled data collection with communication. This created a problem in cases where the sensor node could not connect to the cluster head. In the future, we plan to completely decouple sensing from communication to ensure that we get samples periodically even if we do not send it to the Stargate until the communication can be re-established. **Time synchronization:** In the field deployment, we implemented crude time synchronization by having the Stargate periodically transmit a time beacon through the cluster head. We only needed accuracy in the order of seconds since leak localization was not a goal of the field deployment. In future deployments, we will use a more accurate time sync mechanism that works within a cluster and across clusters; We currently use GPS PPS signals as in our lab setup on the gateway node, and are working on the implementation of an improved version of Vanderbilt's FTSP [26] protocol for the Intel Mote2 sensor node platform. This will enable time stamping of samples across clusters to sub-millisecond accuracy for time delay estimation (Section 3.2).

5. CONCLUSIONS AND FUTURE WORK

We described the PipeNet system that we developed for detecting and localizing leaks and failures in water transmission pipelines.

PipeNet provides a number of noteworthy properties, including: automated detection of leaks and bursts in water transmission pipelines; near-real time operation with few false alarms; applicability to a range of pipe materials; inexpensive to produce, install, and maintain; high-frequency data collection; the ability to differentiate between sensor and system faults and a flexible, reusable data-flow based programming environment. We are optimistic that PipeNet will provide a much needed increase in spatial and temporal resolution of hydraulic, acoustic and water quality data that will improve the ability to understand and monitor large scale water supply and sewer systems.

To fully realize this vision, we are currently developing the second revision of PipeNet based on our experiences from the field and lab deployments. We are using the Intel Mote2 platform [24] which integrates many essential components to enable high performance, energy efficient data processing. Its XScale processor has dynamic voltage and frequency scaling capability to allow applications to balance performance and energy needs by selecting speeds between 13 and 624 MHz. In addition, the processor includes a DSP co-processor to accelerate common data analysis primitives (e.g FFT, compression) thereby greatly improving performance and energy efficiency. This performance advantage will allow us to carry out the analysis and data reduction in real-time, thus reducing storage and power. Finally the Intel Mote 2 includes 32 MB of SDRAM and Flash enabling the decoupling of data collection and communication and a richer peripheral support which will provide higher data acquisition rates and improve sensor integration. We believe this new hardware platform, combined with our next generation in-network processing system (WaveScope [9]) will substantially alter the way water utilities monitor and maintain their pipeline infrastructure.

6. ACKNOWLEDGMENTS

The experimental work was supported by the Cambridge-MIT Institute (CMI), Principal Investigator: Professor Andrew Whittle (MIT). The authors are especially grateful to John Sullivan (Chief Engineer, BWSC) for authorizing the field trial and to Paul Canavan and Tom Bernier (BWSC) for their tremendous help during the installation. In addition, we would like to thank the Intel Mote SRP team for providing the WSN hardware and software components and participating in the field deployment.

7. REFERENCES

- [1] EPA. *Drinking Water Infrastructure Needs Survey and Assessment: Third Report to Congress*. EPA 816-R-05-001, 2005.
- [2] Friedman, M., Radder, L., Harrison, S., Howie, D., Britton, M., Boyd, G., Wang, H., Gullick, R., LeChevallier, M., Wood, D., and Funk, J. *Verification and Control of Pressure Transients and Intrusion in Distribution Systems [Project #2686]* ISBN: 1843398966 AwwaRF Report Series, 2005.
- [3] Stoianov, I., Maksimovic, C. and Graham, N.J.D. *Designing a Continuous Monitoring System for Transmission Pipelines*, In Proceedings CCWI 2003 Advances in Water Supply Management Conference, London, (London, UK), 2003.
- [4] EPA.. *Combined Sewer Overflows Demographics*. (http://cfpub1.epa.gov/npdes/cso/demo.cfm?program_id=5). Accessed October, 2006.
- [5] Nachman, L.Kling, R., Adler, R., Huang, J., and Hummel, V., *The INTEL® Mote Platform: A BLUETOOTH-Based Sensor Network For Industrial Monitoring*. In Proceedings of IPSN, 2005.
- [6] Crossbow, Inc., Stargate gateway, <http://www.xbow.com/Products/productsdetails.aspx?sid=8>.
- [7] BWSC field trial web interface: <http://db.csail.mit.edu/dcnuil/>
- [8] Tokmouline, T. *A Signal Oriented Stream Processing System for Pipeline Monitoring*, MIT Master's Thesis, 2006.
- [9] Girod, L., Jamieson, K., Mei, Y., Newton, R., Rost, S., Thiagarajan, A., Balakrishnan, H., Madden, S. *WaveScope: A Signal-Oriented Data Stream Management System*, CIDR 2007.
- [10] Stoianov, I., Dellow, D., Maksimovic, C., and N.J.D. Graham. *Field Validation of the Application of Hydraulic Transients for Leak Detection in Transmission Pipelines*. In The Proceedings of CCWI 2003 Advances in Water Supply Management Conference, London, UK, September 2003.
- [11] Covas, D.; Stoianov, I.; Ramos, H.; Graham, N.; Maksimovic, C. *Dissipation of pressure surges in water pipeline systems*. In Proceedings Pumps, electromechanical devices and systems applied to urban water management, PEDS 2003, Valencia, Spain, 2003.
- [12] Covas, D.; Stoianov, I.; Mano, J. F.; Ramos, H.; Graham, N.; Maksimovic, C. *The dynamic effect of pipe-wall viscoelasticity in hydraulic transients. Part I-experimental analysis and creep characterization*. JOURNAL OF HYDRAULIC RESEARCH. VOL 42; 5, pp: 516-530, 2004.
- [13] Hunaidi, O. and Chu, W.T. *Acoustical characteristics of leak signals in water distribution pipes*, Applied Acoustics 58, pp 235 –254, 1999.
- [14] Hunaidi, O. Chu, W.T., Wang, A., and W. Guan, *Detecting leaks in plastic water distribution pipes*, Journal of the American Water Works Association 92 (2), 2000.
- [15] Fuchs, H.V. and Riehle, R. *Ten years of experience with leak detection by acoustic signal analysis*, Applied Acoustics 33 pp 119, 1991.
- [16] Charles H. Knapp and G. Clifford Carter, "The generalized correlation method for estimation of time delay", *IEEE Tran. on ASSP*, vol. ASSP-24, no.4, pp.320-327, 1976.
- [17] Etter, D and Samuel D. Stearns, *Adaptive estimation of time delays in sampled data systems*, *IEEE Tran. on ASSP*, vol. ASSP-29, no.3, pp.582-587, 1981.
- [18] Widrow, B. *Adaptive filters*, Aspects of Network and System Theory, R. Kalman and N. DeClaris, New York: Holt, Rinehart and Winston, 1971, pp. 563-587.
- [19] Long, R., Lowe, M., and Cawley, P. *Attenuation characteristics of the fundamental modes that propagate in buried iron water pipes*, Ultrasonics, Vol 41, pp 509-519, 2003.
- [20] Gao, Y., Brennan, M.J., Joseph, P.F., Muggleton, J.M. and O. Hunaidi, "A model of the correlation function of leak noise in buried plastic pipes." *Journal of Sound and Vibration* 277, pp133 –148, 2004.
- [21] Kim, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., Pakzad, S. *Structural Health Monitoring*", Presentation, CENTS Retreat. http://www.cs.berkeley.edu/~binetude/ggb/doc/050113_CENTS_Retreat.ppt
- [22] Boston Globe, *Untreated Sewage released into bay*. Cover page article on the 17th October, 2005:
- [23] Muggleton, J.M. and Brennan, M.J. *Axisymmetric wave propagation in buried, fluid-filled pipes: Effects of discontinuities* *Journal of Sound and Vibration*. 281, 2005.
- [24] Intel Mote 2 platform, <http://platformx.sourceforge.net>
- [25] Krishnamurthy, L., Adler, R., Buonadonna, P., Chhabra, J., Flanigan, M., Kushalnagar, N., Nachman, L., Yarvis, M. *Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea*. Sensys 2005
- [26] Miklos Maroti, Branislav Kusy, Gyula Simon, Akos Ledeczki. *The flooding time synchronization protocol*. SenSys 2004