Embedded Damage Detection in Water Pipelines Using Wireless Sensor Networks

Hadil Mustafa
Department of Electrical Engineering
and Computer Science
University of California, Irvine
Irvine, CA, 92697–2625
Email: hmustafa@uci.edu

Pai H. Chou
Department of Electrical Engineering
and Computer Science
University of California, Irvine
Irvine, CA, 92697–2625
Email: phchou@uci.edu

***Abstract*—This paper proposes a distributed real-time detection algorithm for detecting rupture events in water pipelines noninvasively. The purpose is to suppress not only unnecessary transmission but also local processing in order to save power without sacrificing sensitivity or specificity of the events of interest. All these goals are accomplished by adaptive thresholding, a cascaded wake-up chain, local processing, and aggressive power management. Experimental results show that the proposed algorithm achieves high sensitivity and high specificity while reducing the total energy consumption significantly.**

***Index Terms*—Water pipeline monitoring, ruptures, wireless sensor network.**

I. Introduction

The aging and deteriorating infrastructure is one of the greatest challenges facing water systems in the United states. The EPA (Environmental Protection Agency) estimated a total need of US$324.9 billion over the next 20 years to be used for water infrastructure assessment in their *2007 Drinking Water Infrastructure Needs Survey and Assessment* [1]. The 39 billion dollars estimated for California's need represents about 12 percent of the national need. The DWR (Department of Water Resources) [2] made several recommendations to direct effort to upgrade, improve and enhance the security and emergency response capability of the water infrastructure in order to maintain a reliable supply and delivery of drinking water in the case of damage caused by natural disasters or deteriorating pipelines.

Non-invasive monitoring of pressurized water pipelines is desirable due to cost and practical considerations. Since change in pipe vibration can be attributed to the sudden change in the water pressure caused by rupture or breakage event, non-invasive monitoring can be achieved by measuring the vibration on the pipe surface at various joints. Identifying the location of failure quickly and easily can be crucial in most situations, though it requires continuous monitoring of the system. Challenges associated with monitoring such systems include quality of data sensing, timely and reliable communication from the field to control centers, accurate data analysis, and high power efficiency. The ideal system should be able to continuously monitor the pipes and report upnormal behavior back to the control center while maintaining low power consumption, yet power can not be saved with continuous monitoring. Up-to-date, a tradeoff had be to made between missing events , through using sparse monitoring, and saving power.

This paper describes an energy-efficient method for real-time, in-field monitoring and condition assessment of utility water distribution systems, particularly during and after natural disasters. What makes our system unique is its ability to monitor pipes at all times, while maintaining low power consumption. The system consists of multiple wireless communication units and high-precision sensor nodes that can be deployed non-invasively on fresh-water and sewage-water pipelines. Multiple systems work together to collect acceleration data to be analyzed in the field or transmitted back to the laboratory in real time for post-processing.

II. Background and Related work

A. Background

Online damage detection using WSNs has emerged in recent years as a promising technique to monitor the health state of civil structures such as bridges, buildings, and dams, and also in monitoring water and gas pipelines.

A number of monitoring techniques have been applied in the gas and oil industries to monitor gas pipelines and detect pipe breaks. Most of the applied pipelines monitoring was implemented using invasive methods to measure pressure and flow [3]. Ruptures and breaks in pipelines induce a negative pressure wave that travels in both direction away from the failure point. Measuring and sampling pressure at both ends of the water pipeline can be used to detect rupture events. The performance of such monitoring systems can be dramatically improved by installing more measuring points, but it requires a great deal of instrumentation for continuous monitoring. Water pipelines, as opposed to oil and gas pipelines, are not well-instrumented to allow invasive installations, in addition to the fact that the water utilities budget does not allow for expensive instrumentation. The research has been shifted toward the usage of non-invasive methods to monitor fresh water and sewage pipelines.

Ruptures and leaks in water pipelines manifest themselves as high-amplitude noise, which can be used to detect such

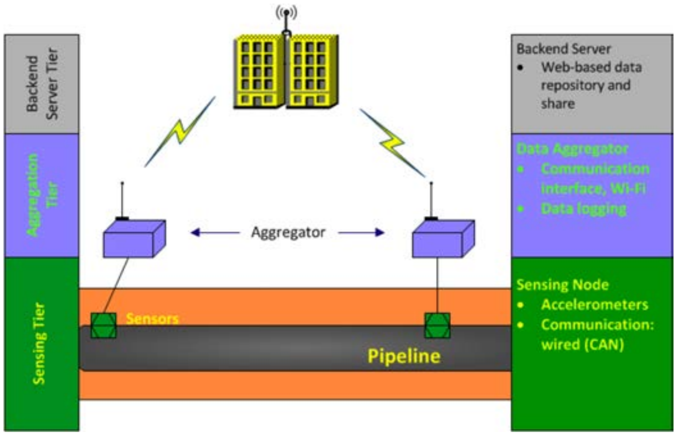


Fig. 1. Original PipeTECT Architecture

events. Previous proposed and implemented WSNs for monitoring water pipelines varied in their sensing techniques, mathematical formulation, data acquisition methods, and data processing algorithms. In most cases, the wireless sensing platform is mainly equipped with sensing, communication, and computation units. The communication unit enables the sensing platform to transmit data wirelessly without the usage of expensive coaxial cables. The computaion unit, if present, is utilized to process data locally and make decisions on the state of the monitored structure or pipes. The sensing unit is equipped with different kinds of sensors, depending on the monitored feature, such as acoustic or vibration sensors, temperature sensors, or even Lead Ziroconate Titanate (PZT) sensors [4].

In our previous work [5], [6], we introduced a wireless sensor platform, PipeTECT, a smart wireless sensor system based on MEMS accelerometers that can be deployed non-invasivly on fresh water or sewage water pipelines. The latest version of PipeTECT encapsulates sensors, processors and communication modules with high performance and expandability features. Fig. 1 shows a block diagram of the original monitoring system. The system consists of three main tiers: the sensing tier, the aggregation tier, and the back-end server tier. The sensing tier consists of the sensing node and low-complexity microcontroller unit (MCU) in addition to a signal converter. The aggregation tier consists of a wireless communication unit and a low-power, low-complexity MCU. The sensed data is transfered from the sensing unit to the aggregation unit through the Controller Area Network (CAN) interface and then continuously streamed back to the server via a wireless connection.

An initial version of the server receives and processes incoming data to determine failure events. Multiple aggregation nodes can aggregate data from multiple sensor nodes, where each sensing node can provide up to 3 channels of acceleration reading (X, Y, and Z axes). At the network level, each aggregation node provides a data stream that can encapsulate several sensing channels. The server saves each data stream in a common data storage for further analysis.

B. Related Work

Monitoring buildings, bridges, dams and pipelines through measuring seismic vibration using WSN has become an active area of research in recent years. The main advantage of using WSN is ease of installation and deployment in addition to lower cost when compared to wired techniques.

Stoianov [7], [8] proposed and evaluated comprehensive method to detect both leaks and breakage in water pipelines using a combination of pressure transient, flow, and acoustic vibration signals. Note that pressure and flow require invasive sensing for detecting breaks and ruptures, while acoustic and vibration are non-invasive. The use of invasive snsors limits the installation to areas with an outlet. The application of non-invasive method used vibration sensors to detect small leaks and relied on continuous sampling for short periods of time to save energy. Hence, leakage events are not time critical. The non-invasive method used FFT analysis to determine leak events in the monitored pipes. Although the analysis were able to detect small leaks in water pipes, it was computationally expensive ($O(N\log(N))$ steps to compute the frequency spectrum for N samples) and required offline data evaluation.

Lynch in [9] proposed embedding a damage detection algorithm in wireless sensor network to lower energy consumption in SHM and overcome the need for continuous streaming of data. A computational core was incorporated to execute engineering analysis and reduce the size of the transmitted data by transmitting only important aspects of the data. Reducing the size of the transmitted data is an effective way to save bandwidth and energy consumption, but it does not overcome the problem imposed by the need for continuous sampling at high rates in applications such as water pipeline monitoring for disastrous events. In addition, the embedded algorithm used in structural-health monitoring relies on the assumption that the measured data can be modeled as a stationary process, i.e., a process where the variance and autocorrelation structure do not change over time. Unfortunately, it is not applicable to water pipeline monitoring because vibration in pipes is highly affected by exotic factors and varies throughout the day.

Geof [10] proposed an event detection algorithm to suppress continuous transmission and save energy in a volcano monitoring network. The event detection algorithm computes two exponentially weighted moving average (EWMA) using two different gain settings. Through collaboration between all sensors in the network, the event is confirmed if 30% of the nodes reported average deviation. While applying a voting scheme to detect events works well for dense WSNs, it is not applicable to water pipe monitoring, where the number of installed sensors are much more limited due to accessibility and wide coverage area. Most of the pipes are buried under ground and access is only possible through manholes and fire hydrants.

Another way to lower energy consumption and reduce the amount of data is to take into account the nature of the monitored application. For example, in [11] monitoring a railway bridges was enabled only when trains are passing.

The network was put to sleep most of the time and was active for short periods of times throughout the day.

In this paper, we present a system architecture for monitoring rupture and breakage events in water pipelines by continuous monitoring while incurring low energy consumption. Our system combines a tiered system architecture, cascaded wake-up hierarchy, and a hybrid local event detection algorithm that work together to monitor the pipes at all times. It extracts and transmits only the important aspects of the data.

The rest of this paper is organized as follows. Section III discusses the problems associated with monitoring water pipelines and outlines the problem statement, followed by section IV to introduce our technical approach. In Section V we present our damage detection algorithm. Section VI describes the experimental setup and field deployment, and Section VII presents the results with an analysis. Section VIII concludes the paper with a direction for future research.

III. Problem Statement and Contributions

The problem statement can be stated in terms of the functional requirements, objective functions, and constraints.

A. Functional Requirements

The functional requirement is to detect rupture events on water pipes by in-field processing. More specifically, the data aggregator collects raw data from the sensing nodes and performs processing to decide whether a rupture event has occurred. If so, it is to report the event back to the server tier immediately. To do this, it is to process the acceleration data sampled at 1000 Hz on the pipe surface. It should report all true-positive events and should not report false-negative ones.

The data collected from our system can fairly be described as semi-infinite stream of values, which can be formally represented by a discrete sequence of numbers $\langle x_1, x_2, x_3, \ldots, x_n \rangle$, each representing the acceleration measured at specific time. The limited processing and memory resources in the sensor networks makes it impossible to store every measured value. On the other hand, the limited bandwidth makes it rather expensive to transmit all the measurements. Therefore, these limitations imply the need for certain trade-offs: it is impossible to store everything or transmit everything, and furthermore we want to more fully utilize the available resources. This problem can be modeled as a continuous query processing problem.

We are looking for an adaptive algorithm that can identify a pattern and requires no previous knowledge or human guidance. In water pipelines monitoring, the vibration of the pipes are affected by many exotic effects. Therefore, it might be impossible to guide the sensors in identifying failure events due to the large volume of data and limited communication. In detail, the main requirements of the detection algorithm can be outlined as follows:

- Succinct model: the algorithm should be able to capture the disastrous event in real time, using limited resources.
- Unsupervised model: very little to no human intervention.
- Streaming limitations: we cannot afford to transmit back all the data, yet we still need the ability to store and report back any values that deviate too much from the standard deviation of the system.

B. Objectives

The primary objective is to maximize both sensitivity and specificity of the event detection method. The secondary objective is to minimize the total energy consumption, or equivalently the average power consumption of the sensing tier and the aggregation tier.

Sensitivity and specificity are metrics for quantifying the accuracy of sensors in general. They are defined as

$$\text{sensitivity} = \frac{TP}{TP+FN} \tag{1}$$

$$\text{specificity} = \frac{TN}{TN+FP} \tag{2}$$

where TP, TN, FP, and FN stand for true positive, true negative, false positive, and false negative, respectively. The sensitivity objective means that whenever a true rupture event occurs, the system should not miss it, or else some rupture events may go unreported. On the other hand, the specificity objective means that whenever an event that is not a rupture event occurs, the system should filter it out, or else the system may suffer from too many false alarms.

The objective to minimize energy consumption can be further refined. On the sensor, this may mean to suppress sampling at the high rate (1000 Hz) whenever possible, since the ADC and signal conditioning subsystem consumes a significant amount of power. On the data aggregator, this may mean to minimize the uplink transmission of data with low information content. Of course, if the local processing ends up consuming more energy, then the data aggregator should just transmit raw data.

C. Contributions

In this paper we introduce a novel hybrid local detection algorithm that can be embedded in sensor nodes to detect and report abnormal behavior in the monitored platform in real-time. Through combining a hardware thresholding detection technique with a fundamental statistical analysis, our system can achieve the highest possible sensitivity and scored 80% savings in power consumption. We also introduced a combined architecture of low-power and high-performance MCUs utilized together in a master/slave model to detect and analyze the measured data before transmission. The contributions of this paper are as follows:

- We develop a hybrid on-site event detection algorithm based on threshold detection and statistical analysis.
- We introduce a cascaded wake-up hierarchy utilized to lower energy consumption.
- We validate the correctness, sensitivity and specificity of our model using real-time measurements obtained from field deployment.

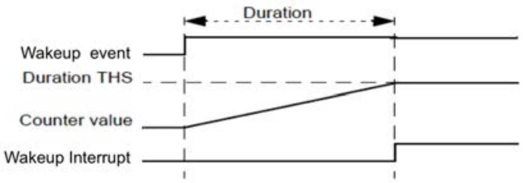


Fig. 2. Threshold violation interrupt

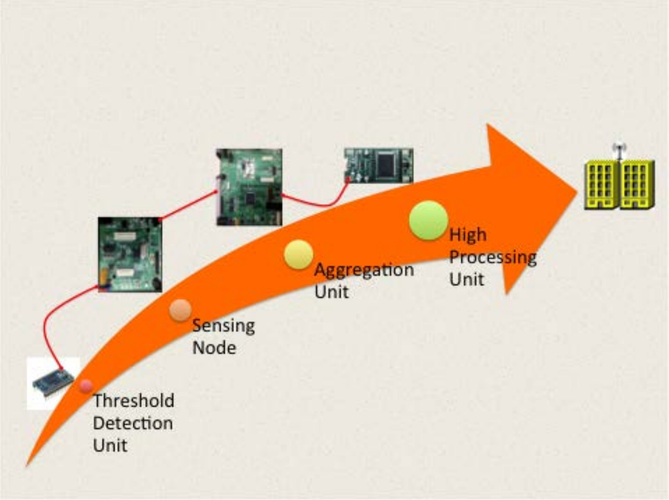


Fig. 3. Cascaded Wake-up Hierarchy

IV. Technical Approach

As mentioned in Section II, the original system architecture had three main tiers: *sensing* tier, *aggregation* tier, and *server* tier. Our approach is to energy-efficient pipeline monitoring is augment the existing PipeTECT system with new modules at both the sensing and the data aggregation tiers. Together, they minimize energy consumption without missing events.

A. Enhanced Data Aggregation Tier

The original data aggregator contains a medium-complexity MCU, the wired or wireless network interfaces, and a data storage module (e.g., flash memory card) for storing and transmitting the data back to the back-end server for analysis. We enhanced the aggregator with a powerful MCU capable of running the analysis algorithm in the field on the collected data, so that it can suppress transmission of data with low information content most of the time. The new MCU is added as a daughter card so that it can be power managed by the existing medium-complexity MCU.

B. Enhanced Sensing Tier

The original sensing node contains a low-complexity MCU, an analog-to-digital converter (ADC) with signal conditioning functions, and a wired bus interface. We added a new module with a threshold detection unit so that the rest of the system can be kept in very low power mode. This added unit enables the cascaded wake-up hierarchy to dramatically reduce the energy consumption without missing events. Fig. 4. The threshold detection unit, samples at lower sampling rate and keeps monitoring the system for any deviation from a preset threshold value throughout the day.

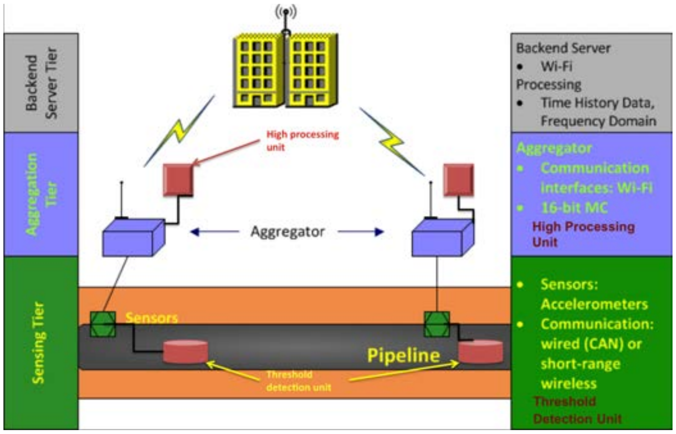


Fig. 4. A block diagram of the monitoring system

C. Wake-up Hierarchy

The cascaded wake up hierarchy starts by the threshold detection unit which monitors the pipes at all times for any deviation off of a predefined threshold value. Upon deviation detection, the threshold unit initializes a wake-up interrupt signal to start the sensing node. Sampling at a higher rate starts on the sensing node and a ripple of wake-up interrupt signals travels from the sensing unit through the aggregation unit, ending at the high computational unit as shown in Fig. 3.

Transmission to the back-end server is only initialized if the failure event is confirmed by the system on-site. Applying the described wake-up hierarchy aided in saving over 80% of the consumed energy, hence all but the threshold detection unit will be in sleep mode most of the time.

D. Two-Tier Event Detection

To achieve these requirements, we propose a hybrid event detection algorithm applied in the lower two tiers, a *threshold hardware based* detector in the sensing tier, and a *statistical based* detector in the aggregation tier. Fig. 5 shows a block diagram of the proposed detection method. The next section describes the algorithm in detail.

V. Hybrid Damage Detection Algorithm

The detection algorithm runs in two stages: *threshold and median estimation* at startup, and *embedded damage detection* at run time.

A. Threshold and Median Estimation

Estimation of the threshold and median values is a crucial step, where it highly affects the sensitivity of the system. For instance, choosing a high threshold value increase the possibility of missing true events related to damage, while lowering the threshold value increases the number of false positives, as the system will falsely report damage. As mentioned, vibration on water pipes are affected by many possible events such as

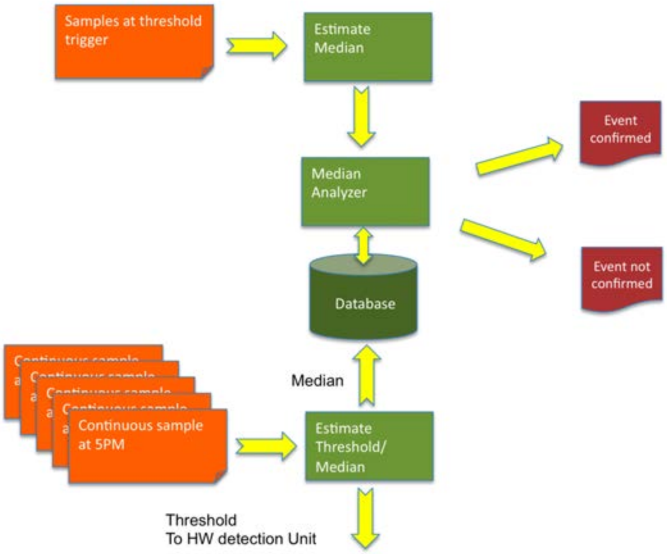


Fig. 5. A block diagram of the two-tier detection method.

```
Algorithm 1: Threshold estimation, THRESHOLD($A,n$)
for  $k = n/2$ to 1 do
    call ShiftDown(k, a, n);
for $k = n$ to 2 do
    $v \leftarrow a[1]$ ;
    $a[1] \leftarrow a[k]$;
    call ShiftDown(1, a, $k-1$);
    $a[k] \leftarrow v$;
procedure $ShiftDown(k, a[], N)$;
  $v \leftarrow a[k]$;
  while $2k = N$ do
      $j \leftarrow 2k$;
      if $j < N \wedge a[j] < a[j+1]$ then
          $++j$
      if  $v \geq a[j]$ then
          break
      $a[k] \leftarrow a[j]$;
      $k \leftarrow j$;
  $a[k] \leftarrow v$;
```

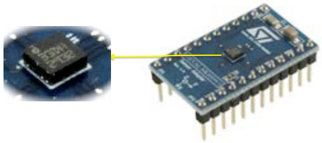


Fig. 6. Threshold Detection Unit

operating pumps, passing trucks, etc. Therefore, the estimation of these values should be associated by the time of the day, since no one single threshold value or median values can be representative at all times. The estimation process runs as follows:

1. continuous sampling during pre-specified times (5AM, 9AM, 12PM, etc).
2. Evaluate and store threshold value T_t associated with t time of the day in a set of data Y sampled at time t where,

$$Y = \langle (y_i) \in R : i = 1....N \rangle : y_i \leq y_{i+1} \leq ....y_{N-1} \leq y_N \tag{3}$$

   The threshold value $T_t \in Y$ will be estimated such as:

$$T_t = Y_n : n = \lfloor .95 * N \rfloor \tag{4}$$

   To optimally evaluate the threshold value on-site, we used in-place sorting algorithm with $O(n \log n)$ complexity. Algorithm 1 shows the pseudocode used to estimate threshold value T in set A with size n [12].

   The obtained optimal threshold value will be sent to the threshold detection unit.
3. Estimate the median m for n set of values where the median y_i for a set Y of size n samples, occurs at

$$i = \lfloor (n+1)/2 \rfloor . \tag{5}$$

4. Populate onsite database with time stamped median values.

B. Embedded Damage Detection

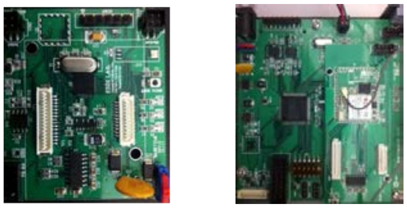
Damage in water pipelines can be identified when the acceleration measured in one window exceeds the optimal threshold value or its median deviates from the expected median. Our damage detection algorithm identifies damage in two steps: a preliminary damage detection based on threshold violation, and a refined detection based on the median deviation. The time stamped threshold values, evaluated at startup, will be stored in the threshold detection unit. As shown in Fig. 2, an interrupt will be triggered when the measured acceleration values, within a time window, rises above the threshold. Once a threshold violation occur, the sensing node starts sampling and evaluating the median values. The obtained values will be compared against the time stamped median, and a damage is confirmed if the deviation exceeds a predefined error ε.

VI. Experimental Setup and Field Deployment

We conducted experiments in several settings, both in the laboratory and in the field. In this section, we provide a description of the hardware used in the experiments and briefly describe one of our field deployments.

A. Hardware Description

1) Sensing Units: The sensing tier consists of a sensing node and a threshold detection unit. The sensing node, shown in Fig. 7, consists of a 4-channel programmable signal converter (QF4A512), an SD1221L-002 MEMS-type accelerometer, and a PIC18LF MCU. The sensing nodes can achieve a stable sampling rate of over 1000 samples per second. The sampled data is sent to the local data aggregator for logging, processing, and transmission. Because RF transmission does not work well underground, and due to the lack of power sources at most sensing locations (i.e., on the exterior of underground pipes), the Controller Area Network (CAN) was



(a) Sensing Unit (b) Aggregation Unit

Fig. 7. Sensing and Aggregation Units

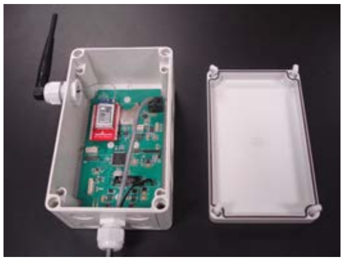


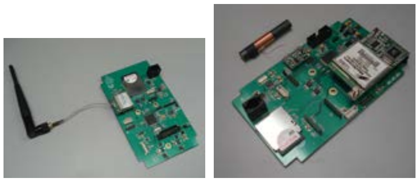
Fig. 8. The Aggregation Unit with WiFi attached

chosen for providing the data link and power to the sensing nodes over a wired interface. Multiple sensing nodes can be daisy-chained together and work as relay points to one aggregation unit. The threshold detection unit shown in Fig. 6 is a triaxial linear accelerometer with an output data rate of up to 400 Hz and a threshold detection capability. The threshold unit is programmed to provide the interrupt signal to the sensing unit, via Serial Peripheral Interface (SPI), when a programmable acceleration threshold is exceeded.

2) Aggregation Tier Hardware: The aggregation tier consists of a data aggregation unit and a high computational MCU.

The aggregation unit, shown in Fig. 7(b), uses the MSP430 16-bit ultra-low-power MCU with 256KB flash, 16KB RAM, and a 12-bit ADC. It does not contain any sensing devices; instead, it contains a Micro-SD card for data logging and several wired and wireless interfaces. It connects to the sensing units via the CAN bus for collecting sensing data, and it also powers all sensing nodes on the attached CAN bus. It has the options of (1) logging data to an on-board, removable Secure Digital (SD) flash memory card, (2) transmitting the data over one of the wireless interfaces, which may be Wi-Fi (by default, as shown in Fig. 8), XBee (up to 1 km), or XTend (up to 64 km) as shown in Fig. 9, or (3) transmit the data to off-board processing unit through SPI for further analysis.

The Computational MCU: An ARM Cortex TMS570LS high-performance 32-bit MCU was selected for data processing and analysis with 1MB flash with ECC, and 160 KB RAM with a fast clock rate of 140 MHz, shown in Fig. 10. The MCU is equipped with three multi-buffered Serial Peripheral Interface (mibSPI) for communicating with the aggregation node. The MCU draws 10mA in sleep mode and 220mA in active mode. Operation of the MCU will be under control of the aggregation node, where sleep, wake-up, and data exchange commands will be communicated through SPI.



(a) XBee (b) Xstream

Fig. 9. Wireless Communication Units

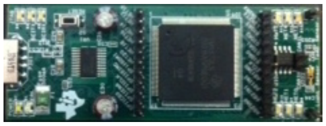


Fig. 10. High Computational Unit

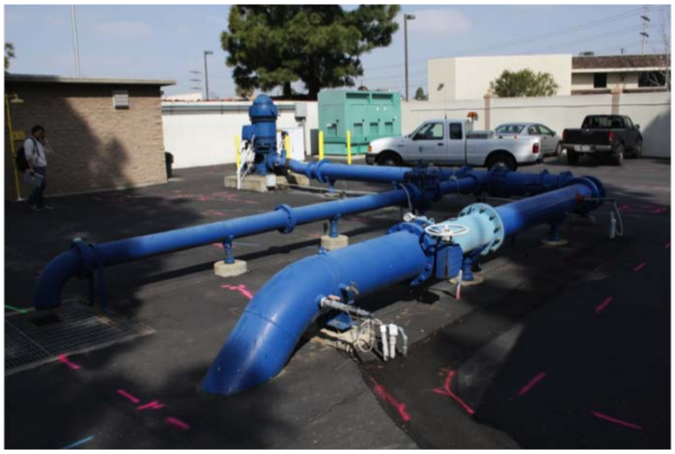


Fig. 11. Experimental Setup Showing PACE Metal Water Pipe

B. Field Deployment

The types of water pipelines that our system can monitor may be pressurized vs. gravity pulled; buried underground vs. above ground; accessible in a manhole, a pump station, a vault, or along another manmade structure such as a bridge; with availability of utility power, energy harvesting, or battery. For the purpose of this study, we analyzed the data collected at the Pacific Advanced Civil Engineering (PACE) firm, located in Santa Ana, CA, USA. We installed the accelerometers on the exterior of the pipes without invasive modifications to the pipes. The installation was relatively easy, as it entailed mainly gluing the sensor (in our case, the sensing node) to the pipe surface with a hot glue gun without compromising the structural integrity of the pipe.

Rupture was simulated by opening a control valve of a metal pipeline system. As shown in Fig. 11, most parts of the pipe were underground but the instrumented section was above ground.

A pressure gauge had already been installed in this section, which enabled us to correlate vibration with pressure change. Three sensing nodes were deployed at

i) upstream 7m away from the control valve,

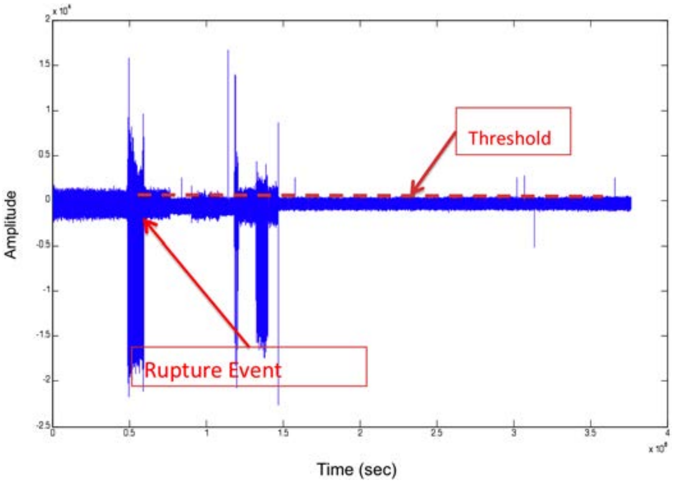


Fig. 12. Acceleration data during rupture emulation test

ii) upstream near the control valve, and
iii) downstream near the control valve.

The sensing node measured the acceleration change in vertical direction. The sampling frequency was set to 1 ksps. The Nyquist frequency (500Hz) was thought to be enough to cover dominant frequency ranges of metal pipes. Three rupture emulations took place during this deployment.

VII. Results and Analysis

As mentioned in Section V, the system runs in two stages: threshold/median estimation and embedded damage detection. We analyzed the data collected through 24 hours operation to estimate the threshold and median values and to populate the database. The estimated values were tested against the data obtained from running rupture emulation tests. The test took place at the PACE site where three rupture emulations took place during a 10-hour operation period, as shwon in Fig. 12. In this section, we analyze the threshold/median estimation results, damage detection results, energy consumption savings, and present the model accuracy.

A. *Threshold/median estimation*

Through monitoring and analyzing the data collected during normal operation for a 24-hour period, we recorded the following observations:

- As shown in Fig. 13, the variation of the threshold values were hardly affected by variable working conditions (operating pumps, heavy usage, etc) and street activities (passing cars, etc) throughout the day. The variations in the threshold values were less than 5% (compared to 22% variance in median values). We can safely select one threshold value that will be suitable for the preliminary threshold violation detection throughout the day.
- The median estimation was more sensitive to varying working condition and time of the day than threshold values, Fig. 14.

  By the end of this stage, we have selected an optimal threshold value and several time-stamped median values, each corresponds to specific time of the day, to be used in the on-time event detection. Relaxing the time dependency of the selected threshold values saved energy and memory in the sensing tier. We needed to store one value in the threshold detection unit that will be used throughout the day as opposed to storing several values corresponding to different times of the day. It also eliminated the need to periodically wake-up the sensing node to update the reference threshold.

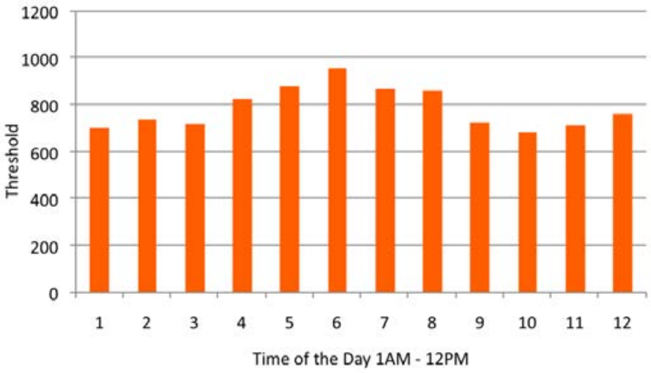


Fig. 13. Variation of threshold values with operation conditions

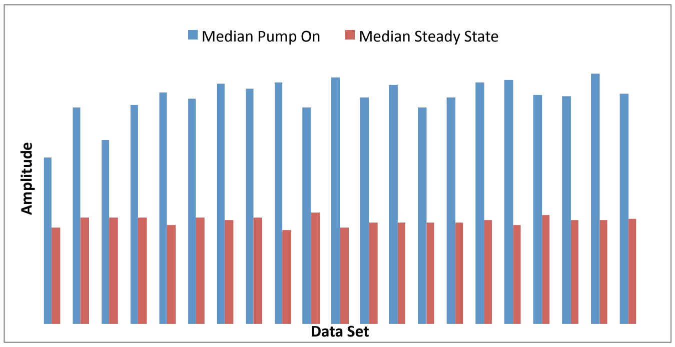


Fig. 14. Variation of median with working conditions

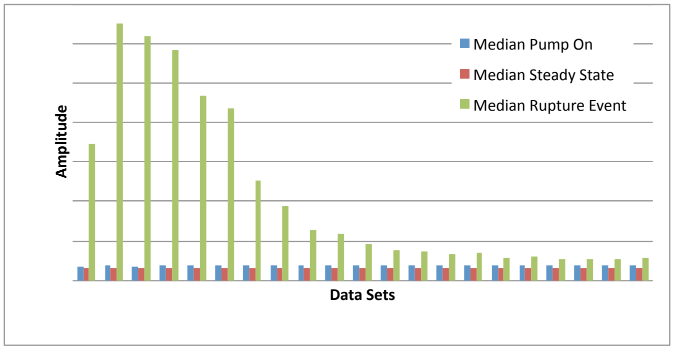
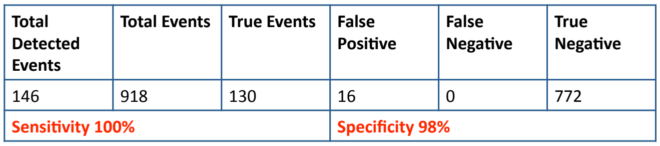


Fig. 15. Median values comparison



Total Detected Events	Total Events	True Events	False Positive	False Negative	True Negative
146	918	130	16	0	772
Sensitivity 100%			Specificity 98%		

Fig. 16. Hybrid event detector accuracy

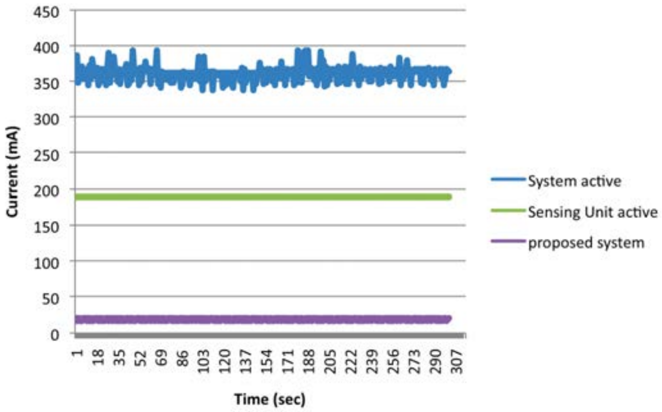


Fig. 17. Current consumption by various units in the system

B. Damage Detection Results and Model Accuracy

The estimated threshold value was used to trigger sampling by the sensing unit once a deviation from threshold was detected. The preliminary threshold detection triggered events in 26% of the data. The system estimated the median of the collected samples and the values were compared to the corresponding time-stamped median stored in the aggregation tier. Fig. 15 shows the variation of median values with working condition. The median-based detection confirmed the events on 16% of the data. At the end of the rupture emulation test, all rupture events were detected successfully. The system achieved a 100% sensitivity and a 98% specificity. The total analysis is shown in Fig. 16.

C. Energy Consumption

One of the main problems in our previous implementation was the high energy consumption of the system. The high sampling rate by the sensing unit, ADC operation, and Wi-Fi transmission were the main causes behind the high energy consumption. As seen in Fig 17, the original system consumed about 380 mA at 12 V, representing a power supply requirement of 4.56 W while active (both sensing and aggregation units running). The sensing unit alone required 2.4 W while active, which represent over 52% of the total power consumption (200 mA at 12 V). In reality, if we exclude sudden catastrophic events (such as earthquakes or accidental drilling), any changes to the state of the pipes will take place over weeks, months or even years, which justifies keeping most of the components in the monitoring system OFF or in SLEEP mode most the time, while keeping the threshold detection unit ON to continuously monitor the pipes at all times. The threshold detection unit consumes as low as 400 μA at 2.5 V, requiring a power supply of 1 mW. Theoretically, and to put all this into context, in the original PipeTECT system a marine battery with a capacity of 300Ah at 12 V lasted less than 30 days, in the best scenarios, the same battery can last over 80 years using our proposed design.

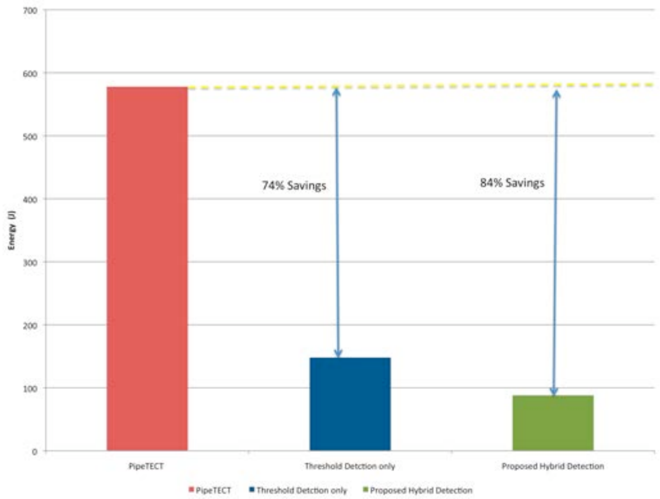


Fig. 18. Energy consumption comparison

We applied further analysis to validate the power efficiency of the proposed design while considering rupture events and abnormal activities using the data obtained from the PACE rupture emulation test. During the experiment, the original system consumed 578 J to sample and transmit over 3M samples of data. while it consumes 148 J if we activated the threshold detection unit and applied the threshold detection algorithm resulting in 74% of savings. Fully applying the hybrid detection algorithm resulted in 84% of savings, as seen in Fig. 18. Although the addition of the high computational unit with median analysis achieved only 10% extra saving in energy consumption over applying the threshold detection, it resulted in minimizing false positive (which may trigger data collection and transmission by the aggregator for false events) and thus improved the systems' specificity from 86% to 98%.

D. Real-Time Requirements

One critical aspect on our design was to make sure the system is capable of reporting rupture events in real-time. The response time to events had to be fully tested and validated. To illustrate the timing evaluation, consider an operational scenario where the sensing tier is triggered to collect a raw time-history record of 4096 points:

- The event detection in the threshold unit takes *700 ms* to detect first event and trigger an interrupt to wake-up the sensing node.
- The sensing node need *100 ms* to wake-up and start all its components (ADC, CAN,etc), at the same time it will create an interrupt to wake-up the aggregator unit.

- Aggregator unit takes less than *1μs* to start its microcontroller and *50 ms* to initialize the WiFi module.
- The sensing unit samples at a rate of 1000 samples/sec, to collect a time-history of 4096 samples it needs *4.096 sec*. It starts transmitting the data directly to the aggregator through CAN interface, the CAN interface has a transmission rate of up to 1Mbps (each sample produced by the A/D converter is represented by 16 bit, 4096 samples results in 65,536 bits).
- Running local analysis on 4096 samples at the aggregator tier took *50ms*
- Transmitting 65,536 bits back to the back-end server takes *142 ms* (WiFi transmit at a rate of 460,800 bits/sec).

The total response and report time needed for the first package to reach the back-end server is less than 6 sec *(700+100+4096+50+142=5138 ms)*.

VIII. Conclusion

This paper introduces an approach to monitoring water pipelines in real-time and detecting rupture events. The system combines a hybrid adaptive detection algorithm with a cascaded wake-up hierarchy to minimize energy consumption without sacrificing the sensitivity or specificity of the detected events. It was found that alternating between ON, OFF and SLEEP mode, in the sensing tiers, and suppress transmission can achieve more than 80% savings in energy consumption. The lower energy consumption resulted from taking advantage of sleep mode that was applied to all but the threshold detection unit. We managed to allow the system to fully operate only when a preliminary event detection was triggered. In addition we successfully applied an adaptive sampling rate, where sampling at a higher rate was only initiated when a threshold violation was detected. A hybrid event detection algorithm was implemented on a low-power threshold-hardware-based detector and a high-performance processing unit in the sensing and aggregator tiers, respectively. We also adapted an aggressive approach to save energy and prolong battery lifetime by allowing transmission of data only when an event was confirmed by the statistical based event detector. Our preliminary evaluations gave promising results where we were able to detect 3 true rupture emulated events that took place during 12 hrs of normal operation. Our evaluation has achieved a high sensitivity of 100% and a satisfying specificity of 98%. Our ongoing work entails incorporating fault tolerance in our sensing system. For future work, we plan to add an emergency recovery scheme in order to recover and eliminate the single point of failure in the system.

Acknowledgment

This study was sponsored by the National Institute of Standards and Technology (NIST) with a Technology Innovation Program (TIP) Grant 080058.

References

[1] EPA (Environmental Protection Agency), “2007 drinking water infrastructure needs survey and assessment: Fourth report to congress,” EPA 816-R-09-001, March 2009.

[2] California Water Plan Update 2009, “DWR (department of water resources),” http://www.waterplan.water.ca.gov/, 2009.

[3] D. Misiunas, J. Vitkovsky, G. Olsson, A. Simpson, and M. Lambert, “Pipeline break detection using pressure transient monitoring,” *Journal of Water Resources Planning and Management*, vol. 131, no. 4, July 1 2005.

[4] Y. Jin and ali Eydgahi., “Monitoring of distributed pipeline systems by wireless sensor networks,” in *Proceedings of the IAJC-IJME international conference*, 2008.

[5] M. Shinozuka, P. Chou, S. Kim, H. R. Kim, E. Yoon, H. Mustafa, D. Karmakar, and S. Pul, “Nondestructive monitoring of a pipe network using a mems-based wireless network,” in *Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security IV*, San Diego, CA USA, March 9 2010.

[6] S. Kim, E. Yoon, T.-C. Chien, H. Mustafa, P. Chou, and M. Shinozuka, “Wireless sensor system for lifeline health monitoring under a disaster event,” in *Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security V*, San Diego, CA USA, March 9 2011.

[7] I. Stoianov, L. Nachman, and S. Madden, “PIPENET: A wireless sensor network for pipeline monitoring,” in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, 2007.

[8] I. Stoianov, L. Nachman, A. Whittle, S. Madden, and R. Kling, “Sensor networks for monitoring water supply and sewer systems: Lessons from boston,” in *Proceedings of the 8th Annual Water Distribution Systems Analysis Symposium*, 2006.

[9] J. P. Lynch, A. Sundararajan, K. H. Law, A. S. Kiremidjian, and E. Carryer, “Embedding damage detection algorithms in a wireless sensing unit for operational power efficiency,” *Journal of Smart Materials and Sturctures*, vol. 800, no. 13, June 1 2004.

[10] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, “Fidelity and yield in a volcano monitoring sensor netwrok,” in *Proceeding of the 7th symposium on Operating systems design and implementation*, Berkeley, CA USA, 2006.

[11] K. Chebrolu, B. Raman, N. Mishra, P. K. Valiveti, and R. Kumar, “Brimon: A sensor network system for railway bridge monitoring,” in *Proceedings of the 6th International Conference on Mobile Systems, applications, and Services (MobiSys)*, 2008.

[12] Thomas H. Cormen and Charles E Leiserson and Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, 2001.