

# Adaptive Edge Analytics for Distributed Networked Control of Water Systems

Sokratis Kartakis, Weiren Yu, Reza Akhavan, and Julie A. McCann

*Department of Computing, Imperial College London, UK*

{s.kartakis13, weiren.yu, reza.akhavan, j.mccann}@imperial.ac.uk

**Abstract**—Over the last decade, there has been a trend where water utility companies aim to make water distribution networks more intelligent in order to improve their quality of service, reduce water waste, minimize maintenance costs etc., by incorporating IoT technologies. Current state of the art solutions use expensive power hungry deployments to monitor and transmit water network states periodically in order to detect anomalous behaviors such as water leakage and bursts. However, more than 97% of water network assets are remote away from power and are often in geographically remote underpopulated areas; facts that make current approaches unsuitable for next generation more dynamic adaptive water networks.

Battery-driven wireless sensor/actuator based solutions are theoretically the perfect choice to support next generation water distribution. In this paper, we present an end-to-end water leak localization system, which exploits edge processing and enables the use of battery-driven sensor nodes. Our system combines a lightweight edge anomaly detection algorithm based on compression rates and an efficient localization algorithm based on graph theory. The edge anomaly detection and localization elements of the systems produce a timely and accurate localization result and reduce the communication by 99% compared to the traditional periodic communication. We evaluated our schemes by deploying non-intrusive sensors measuring vibrational data on a real-world water test rig that have had controlled leakage and burst scenarios implemented.

**Keywords**—IoT, Cyber-Physical Systems, Wireless Sensor Networks, Anomaly Detection, Burst Localization

## I. WATER DISTRIBUTION AND SENSOR NETWORKS

Water security is currently a hot topic. Water demands are not being met in regions of the world; both developed and underdeveloped; where climate change and economic water scarcity are two issues that have the largest impact. The former sees areas of the planet less able to generate enough water for its people, whereas in economic scarcity the principality is unable to build or maintain a water distribution network to continuously meet demands. Drought prone areas such as California in the USA have had severe water restrictions in place for some time. Wet countries, such as the UK, have been experiencing what has been termed wettest droughts over the past few years. Notwithstanding the 7.5bn investment in UK water distribution networks, 3.3bn litres of water were lost per day in 2010 [1].

The use of sensing systems to identify water leakage have been around for some time [2], but their uptake has not been prolific. ICT to support WDN typically consists of remote or online battery-powered telemetry units (data

loggers) that record water data such as flow and pressure, over numbers of minutes, then aggregate this data and send to a server periodically - typically via the mobile phone networks. Contemporary approaches use Wireless Sensor Network (WSN) [2], [3], [4], [5], [6] technologies to monitor the status of the water network and detect leakage or water bursts closer to real-time. The main drawbacks of these approaches are: (a) the analysis of the data takes place off-line, in base-stations or servers meaning that optimal real time decision-making for control would be unrealistic and (b) the sensor nodes require a lot of energy, which places upper bounds on the amounts of data that can be sensed and relayed for analysis. The latter issue particularly impacts on what is important to water companies, leakage localization. Given the cost of digging up roads to fix leaks, timely and accurate determination of the location of a leak is now more important than identifying all leakages. However, to carry out localization high-fidelity sensing and analytics are required to triangulate leakage accurately; current systems are not quite there yet.

In parallel, civil engineers advocate that next generation water networks will not be passive water delivers, but active highly-distributed control systems that route the water intelligently to match demand and route round failures etc. [7]. Such a dynamical system will heavily rely on sensing and actuation and will effect a dual control system of a water distribution network coupled with a smart distributed sensor/actuator network. This will require that both networks interact in a complex cyber-physical way and make use of state-of-the-art low-powered high-precision sensing and processing technologies. It is our premise that state of the art high-precision sensors can help build such a system, but that it is too costly, and not very useful, to send all this higher precision data back to off-line servers for processing. That is, much of the identification of a leak can be carried out on edge devices (i.e. the sensor nodes) and they can collaborate to localize that leak and carry out control. This is a step change from the tradition of sending periodic data to servers to make control decisions, replacing it with a distributed event-based control system.

This paper presents a highly distributed lightweight scheme that combines compression and anomaly detection to identify and locate water leak events and is the first steps to the aforementioned distributed event-based control system. On event detection we only need to send timestamp data

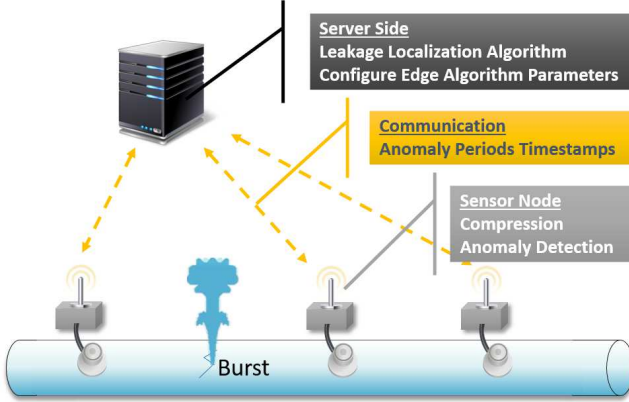


Figure 1: System architecture: sensor nodes, communication, and server side system.

to our novel graph topology-based scheme that recursively examines data from pairs of sensor nodes that detect an anomaly simultaneously to localize the leak. Though inspired by water distribution network leakage management, these algorithms are designed for generic sensing and analytics for other pipe based cyber-physical systems and beyond. Here we focus on the event detection and localization, the complete control function is outside the scope of this paper.

The work presented here has been developed as part of two Smart Water projects (NEC and WISDOM) and we show that we can not only significantly reduce the amount of communications between sensor devices and servers, but also that early transient or event (such as water bursts) detection can run on low-resourced sensor nodes meaning that local control functions can occur with minimal latency. The event detection software uniquely uses compression rates rather than raw data for analysis. The localization scheme demonstrates that it substantially improves positioning accuracy through capturing the water network topology information without sacrificing computational efficiency. The search algorithm combines average distances in the abstractions of the water network topology with differences in arrival times of the sensed anomaly as detected at the sensor locations. As far as we are aware we are the first to provide an edge analytics solution to water network distribution systems and the first to combine this with highly efficient graph processing to localize leaks.

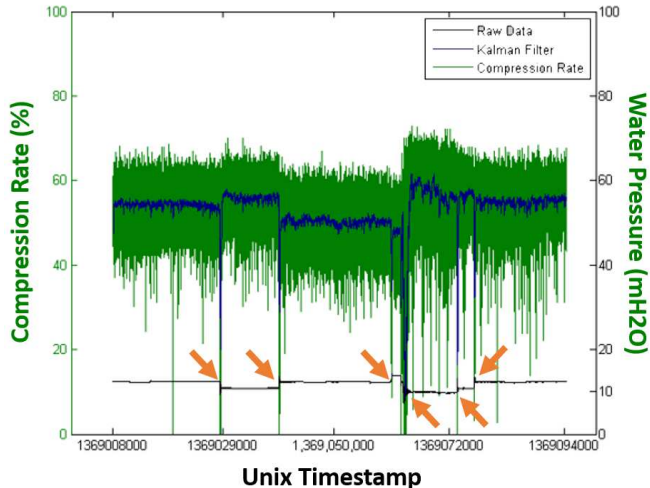
This paper is organized as follows: Sections 2 and 3 presents the edge anomaly detection algorithm and the self-tuning system based on active learning. Section 4 proposes efficient graph-based techniques to effectively localize water burst events. Section 5 demonstrates the experimental environment followed by the empirical evaluation in Section 6. Section 7 surveys the related work and Section 8 concludes the paper.

## II. EDGE-ANALYTICS FOR ANOMALY DETECTION

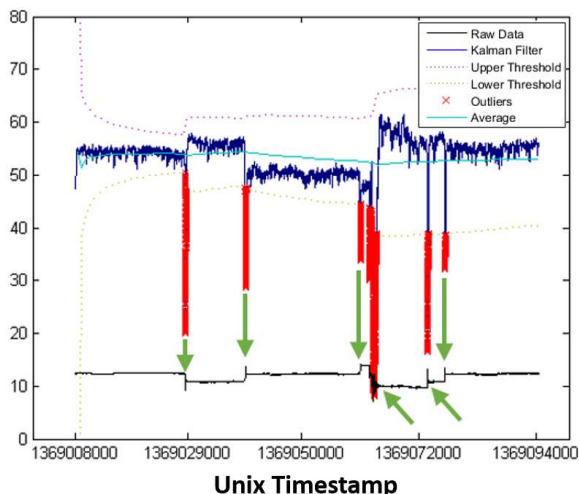
To provide a non-intrusive solution to leakage localization we fit sensor nodes to water pipes. For this project we use the NEC Tokin Ultrahigh-Sensitivity Vibration Sensor that covers a frequency band of 10 to 15 kHz (and acceleration at 0.0001 G) with very low power requirements [8]. Such high fidelity sensors allow us to better explore water network transient phenomena, but the cost of fully transmitting that data is prohibitive using battery powered low-resourced devices. Therefore, each sensor node utilizes lossless compression techniques which reduce the energy cost related to the communication without sacrificing the precision of the data. After extensive evaluation of different lossless compression algorithms [9], miniLZO [10] was selected as the most appropriate algorithm. Our choice takes typical MCU class devices memory and energy constraints into account, e.g. typical ultra-low power MCUs have 64Kbytes memory [11], therefore we limit the compression algorithms working space to 10K.

We now introduce our anomaly detection algorithm that uniquely does not analyse raw data for anomalies that could be leaks, but instead automatically detects significant changes in compression rates in the compressed data and in doing so identifies the timestamps of anomalies that represent leaks [9]. In this paper, we use vibration data - vibration sensors fixed externally to the pipe providing a less intrusive and lower cost sensing solution attractive to water companies. Initially, the input stream is separated into windows of  $w_{stream}$  bytes (i.e. 512 bytes) and for each window, the sensor node applies lossless compression and produces a compression rate value. Figure 2a illustrates both raw pressure data and compression rate (noisy signal). To maximize anomaly detection while minimizing the number of false-positive results, noise is removed from the compression rate stream using a one-dimensional Kalman Filter [12], [13] indicated in Figure 2 and Figure 3 with a blue line.

The use of Kalman filters is motivated by: (a) their support for streaming analysis using only current input measurements (therefore making the solution more memory efficient), (b) they do not require matrix calculations (therefore the solution is more computationally efficient), (c) the ease of the algorithm tuning process, and (d) their implementation simplicity. During the initialization process the parameters which need tuning are the process noise covariance  $q$ , the sensor measurement noise covariance  $r$ , the initial estimated error covariance  $p$  and an initial measurement  $x$  (i.e.  $q = 0.005$ ,  $r = 25$ ,  $p = 0$ , and  $x =$  the first compression rate measurement). Afterwards, for every new compression rate, the Kalman filter algorithm updates these parameters and produces the filtered value of compression rates. After noise removal, the anomalies can be detected accurately as can be seen in Figure 2a the anomalies are presented as great drops (arrows). The anomalies are being detected by using



(a) Raw data, compression rates, and Kalman filter results.



(b) Anomaly detection results

Figure 2: Anomaly detection based on water pressure data.

an adaptive thresholding approach based on the mean and the standard deviation of the compression rate moving average for a predefined window size  $w_{avg}$  (i.e. 128). We use this because it smooths states for easier analysis and reduces threshold computations to window sizes. Specifically, the algorithm computes the moving average of the filtered compression rates ( $x$  values), with the average  $avg$  and the standard deviation  $std$  of the moving average. In every Kalman state update, the algorithm identifies as an anomaly the values which hold the following condition:

$$x > (avg + std * l) \text{ or } x < (avg - std * l)$$

where  $l$  represents the elasticity of anomaly detection (smaller values mean that the system is more sensitive - Figure 4).

As can be observed in Figure 4a, the algorithm suffers

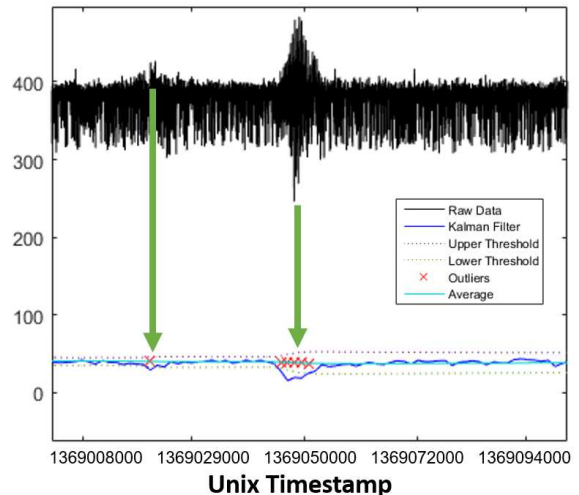
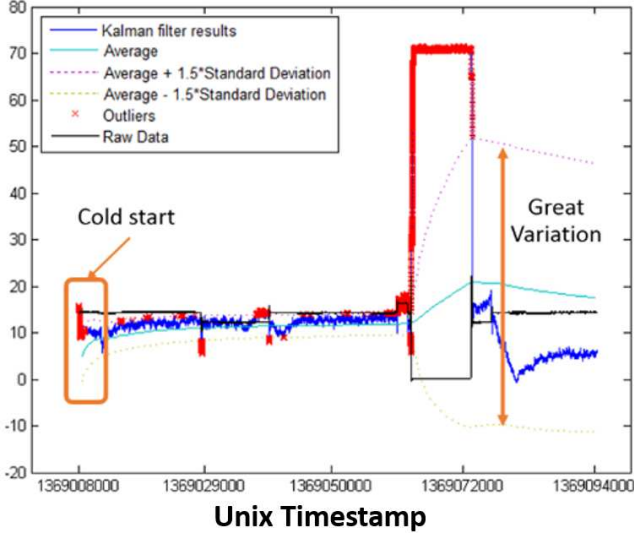


Figure 3: Anomaly detection over vibration data from NEC Token sensors externally fixed to the water pipe

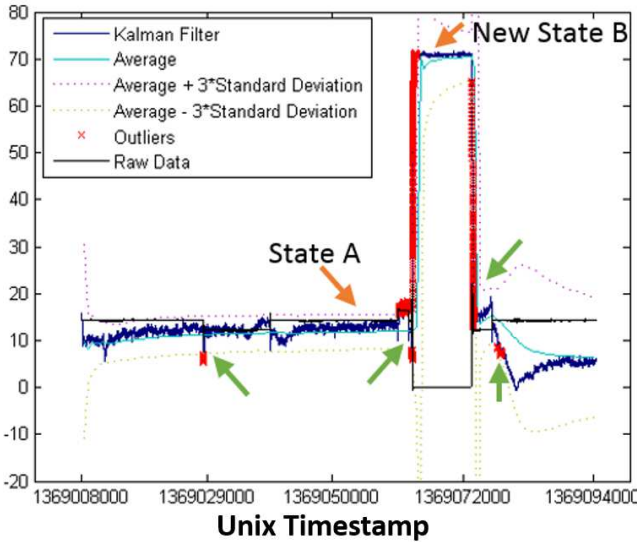
from a cold start effect (it identifies the first values to be outliers because the moving average is yet to be calculated). To solve this problem, the algorithm initializes the  $avg$  and computes  $std$  by using the current compression rate value. Another challenge occurs where there is a significant variation of compression rate data detected (Figure 4a). In that case, because the standard deviation has a high value, this indicates that the algorithm requires more intervals for the moving average calculations to detect the outliers or anomalies. The solution is to reset the values, that is, to initialize the  $avg$  and  $std$ , every time the distance between the thresholds created by the standard deviation become greater than a specific value  $t$  (in our system the  $t = 35$ ). Figure 4b illustrates the anomaly detection results after the reset feature application.

Based on the above analysis, by tuning the input parameters, our algorithm can be applied to any case of high sample rate anomaly detection (i.e. pressure, vibration - see Figure 3) in hardware constrained sensor nodes. However, to maximise the performance of the algorithm, a self-tuning technique is required thus bringing Autonomic properties to the system.

By exploiting this edge anomaly detection algorithm, communication costs can be reduced significantly. The localization system requires only the anomaly detection arrival times from every sensor node. Therefore, each sensor transmits timestamps only instead of raw data, and only whenever an anomaly is observed at the edge. This is in contrast to the traditional periodic sense and send routine of current water network sensing solutions as well as many WSN systems in general. It also better fits modern distributed event-based control systems such as [14].



(a)



(b)

Figure 4: Filtered compression rates and anomaly detection based on (a) pressure data and (b) vibration data (a) Cold Start, Large Variation and inelastic outliers detection ( $l = 1.5$ ), and (b) fixed algorithm results ( $l=3$ ).

### III. OPTIMAL SELF-TUNING VIA ACTIVE LEARNING

By tuning the input parameters, our algorithm can be applied to any case of high sample rate anomaly detection in hardware constrained sensor nodes. Table 1 lists the tuning parameters required by our edge algorithm. In order to optimize the tuning process of the algorithms, we borrow ideas from active learning techniques [15]. Here true anomalies for a single representative training dataset are labelled. For the results that we present here, we applied the active learning idea by asking water data technicians to manually

Table I: Algorithm parameters

Process	Parameters
Input stream split windows size	$w_{stream}$
Kalman Filter initialization	Process noise covariance $q$ Sensor measurement noise covariance $r$ Initial estimated error covariance $p$
Moving average window size	$w_{mavg}$
Threshold Elasticity	$l$
Reset Threshold	$t$

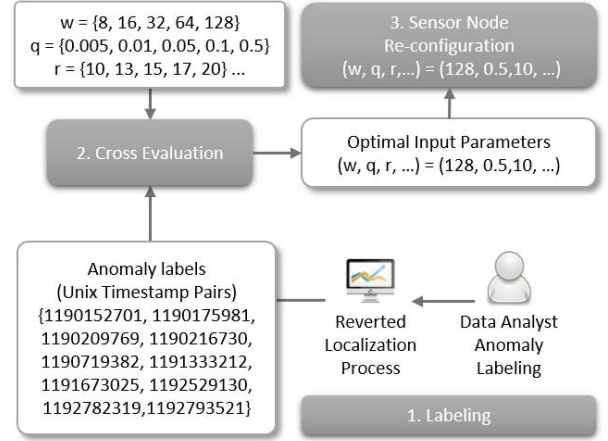


Figure 5: Anomaly detection parameter tuning.

label anomalies on a subset of our evaluation data. Then, we created an offline cross-evaluation system [9], which uses our algorithm and calculates the correct, false/positive ( $FP$ ), and true/negative ( $TN$ ) anomaly detections based on the initial labelling. This establishes the optimal input parameters as the combination that maximizes the distance:

$$Distance D = [Correct - (FP + TN)] Detections$$

Using this, one can imagine that a system would update parameters to re-update the in-node anomaly detection algorithm as the water network evolves. We not only use anomaly identification but also leakage localization in the labelling process; this extends our prior work significantly.

### IV. WATER BURST EVENT LOCALIZATION

In this section, efficient graph-based techniques are used to efficiently and accurately localize the water burst event. We first devise a novel graph topology-based measure that can quantify the “average length” between every two sensor locations, and then define our search method to localize burst events.

#### A. Graph Topology-Based Measures

A water network can be modelled as an attributed graph  $G = (V_J \cup V_S, E, A)$ , where  $V_J$  is a vertex set of pipe junctions,  $V_S$  is a vertex set of deployed sensor locations,  $E$  denotes an edge set of pipe sections connecting two vertices, and  $A$  carries the length of each pipe section.

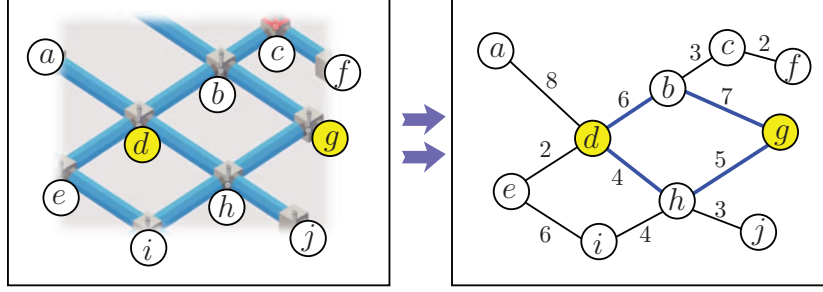


Figure 6: Modelling a water network (left) as a weighted graph (right) based on graph topology

To evaluate the “average length” between every two vertices in a graph  $G$ , let us first introduce the notions of *the distance matrix  $\mathbf{D}$*  and *the adjacency matrix  $\mathbf{A}$* .

*Definition 1:* Given a water network  $G = (V_J \cup V_S, E, A)$  with  $|V| = |V_J| + |V_S|$  vertices and  $|E|$  edges, its distance matrix  $\mathbf{D}$  is a  $|V| \times |V|$  matrix, with its entry  $\mathbf{D}_{u,v}$  being

$$\mathbf{D}_{u,v} = \begin{cases} \text{the length of pipe section } (u,v), & \text{if } (u,v) \in E; \\ 0, & \text{otherwise.} \end{cases}$$

The adjacency matrix of  $G$ , denoted as  $\mathbf{A}$ , is defined by

$$\mathbf{A}_{u,v} = \begin{cases} 1, & \text{if } u \neq v \text{ and } \exists \text{ pipe section } (u,v) \in E; \\ 0, & \text{otherwise.} \end{cases}$$

*Example 1:* Figure 6 depicts the water network  $G$ , whose edge weights are the length of pipe sections. By Definition 1, its distance matrix  $\mathbf{D}$  and adjacency matrix  $\mathbf{A}$  are

$$\mathbf{D} = \begin{matrix} & \begin{matrix} a & b & c & d & e & f & g & h & i & j \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 6 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 8 & 6 & 0 & 0 & 2 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 5 & 0 & 4 & 3 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \end{bmatrix} \end{matrix}$$

$$\mathbf{A} = \begin{matrix} & \begin{matrix} a & b & c & d & e & f & g & h & i & j \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{matrix}$$

Utilizing  $\mathbf{D}$  and  $\mathbf{A}$ , we can determine the “average length” between every two sensor locations on graph  $G$ .

We first introduce a  $|V| \times |V|$  matrix,  $\mathbf{W}^{(d)}$ , whose entry  $[\mathbf{W}^{(d)}]_{(u,v)}$  is the “average length” of all paths with  $d$  hops between vertices  $u$  and  $v$ , that is,

$$[\mathbf{W}^{(d)}]_{(u,v)} = \frac{\text{the sum of the pipe section lengths over all paths with } d \text{ hops between vertices } u \text{ and } v}{\text{the number of the paths with } d \text{ hops between vertices } u \text{ and } v}. \quad (1)$$

To obtain the denominator of Equation (1), we can utilize the property of the power of an adjacency matrix. That is, the  $(u,v)$ -th element of the  $d$ -th power of  $\mathbf{A}$ , that is,  $[\mathbf{A}^d]_{(u,v)}$ , counts the number of the paths with  $d$  hops between vertices  $u$  and  $v$ .

However, evaluation of the nominator of Equation (1) is non-trivial as the power of a distance matrix can only evaluate the *product* (instead of *sum*) of the pipe section lengths over all paths. As an example in Figure 6, to determine the *sum* of the pipe section lengths over all paths with 2 hops between vertices  $d$  and  $g$ , the result of  $[\mathbf{D}^2]_{(d,g)}$  would produce the *product* of the pipe section lengths as follows:

$$\begin{aligned} & [\mathbf{D}^2]_{(d,g)} \\ &= (\text{the } d\text{-th row of } \mathbf{D}) \times (\text{the } g\text{-th column of } \mathbf{D}) \\ &= \begin{bmatrix} 8 & 6 & 0 & 0 & 2 & 0 & 0 & 4 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 7 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 \end{bmatrix}^T \\ &= 6 \times 7 + 5 \times 4 \neq \underbrace{(6+7)}_{d \rightarrow b \rightarrow g} + \underbrace{(5+4)}_{d \rightarrow h \rightarrow g} \end{aligned} \quad (2)$$

We notice that, if the “ $\times$ ” sign in Equation (2) were changed into “+” sign, the result would turn into the more desirable *sum* of the pipe section lengths over all paths ( $d \rightarrow b \rightarrow g$  and  $d \rightarrow h \rightarrow g$ ) with 2 hops between vertices  $d$  and  $g$ . To obtain the correct “+”-based results, we can perhaps take advantage of the power of a distance matrix while changing its “ $\times$ ” sign (in Equation (2)) into “+” sign as well?

To address this, we introduce an element-wise operator  $\exp(*)$ . We construct the element-wise *exponential distance matrix*, denoted as  $\exp(t\mathbf{D})$ , as follows:

$$[\exp(t\mathbf{D})]_{u,v} = \begin{cases} \exp(t\mathbf{D}_{u,v}), & \text{if } \mathbf{D}_{u,v} \neq 0; \\ 0, & \text{if } \mathbf{D}_{u,v} = 0. \end{cases}$$



where  $t \in \mathbb{R}$  denotes an arbitrary scalar.

Intuitively, the matrix  $\exp(t\mathbf{D})$  is formed by replacing every nonzero element in  $\mathbf{D}$ , say  $x$ , with  $e^x$ , and keeping the zero elements of  $\mathbf{D}$  unchanged.

Then, to assess the *sum* of the pipe section lengths over all paths with 2 hops between vertices  $d$  and  $g$ , we compute the  $(d, g)$ -th element of  $(\exp(t\mathbf{D}))^2$ , that is,

$$\begin{aligned} & [(\exp(t\mathbf{D}))^2]_{(d,g)} \\ &= (\text{the } d\text{-th row of } \exp(t\mathbf{D})) \times (\text{the } g\text{-th column of } \exp(t\mathbf{D})) \\ &= \begin{bmatrix} e^{8t} & e^{6t} & 0 & 0 & e^{2t} & 0 & 0 & e^{4t} & 0 & 0 \end{bmatrix} \\ & \quad \begin{bmatrix} 0 & e^{7t} & 0 & 0 & 0 & 0 & 0 & e^{5t} & 0 & 0 \end{bmatrix}^T \\ &= e^{6t} \times e^{7t} + e^{5t} \times e^{4t} = e^{(6+7)t} + e^{(5+4)t} \end{aligned} \quad (3)$$

In contrast to Equation (2), we can see that, by utilizing the operator  $\exp(*)$ , Equation (3) converts all “ $\times$ ” signs into “+” signs. In light of Equation (3), our next step is to find out an “inverse” operator that can map  $e^{(6+7)t} + e^{(5+4)t}$  back into  $(6+7) + (5+4)$ . Our key observation is that

$$\lim_{t \rightarrow 0} \frac{2}{t} \log \left( \frac{1}{2} (e^{xt} + e^{yt}) \right) = x + y \quad (4)$$

Thus, applying the “inverse” operator  $\lim_{t \rightarrow 0} \frac{2}{t} \log \left( \frac{1}{2} (*) \right)$  of Equation (4) into Equation (3) produces

$$\begin{aligned} & \lim_{t \rightarrow 0} \frac{2}{t} \log \left( \frac{1}{2} \left( [(\exp(t\mathbf{D}))^2]_{(d,g)} \right) \right) \\ &= \lim_{t \rightarrow 0} \frac{2}{t} \log \left( \frac{1}{2} (e^{(6+7)t} + e^{(5+4)t}) \right) \\ &= (6+7) + (5+4) \end{aligned} \quad (5)$$

whose result gives the *sum* of the pipe section lengths over all paths ( $d \rightarrow b \rightarrow g$  and  $d \rightarrow h \rightarrow g$ ) with 2 hops between vertices  $d$  and  $g$ .

Equations (3)–(5) provide an effective technique to obtain the nominator of  $[\mathbf{W}^{(d)}]_{(u,v)}$  in Equation (1). To generalize our above result for any arbitrary element of  $(\exp(t\mathbf{D}))^2$ , we need to extend the “inverse” operator in Equation (4):

*Theorem 1:* For any positive integer  $N = 1, 2, \dots$ , the following equation holds:

$$\begin{aligned} \lim_{t \rightarrow 0} \frac{N}{t} \log \left( \frac{1}{N} (e^{x_1 t} + e^{x_2 t} + \dots + e^{x_N t}) \right) \\ = x_1 + x_2 + \dots + x_N. \end{aligned} \quad (6)$$

As a special case of  $N = 2$ , Theorem 1 reduces to the result in Equation (4). Theorem 1 is used for generalizing the result of Equation (3) for any *arbitrary* element of  $(\exp(t\mathbf{D}))^k$ . More specifically, in our aforementioned example, we choose Equation (4) (that is,  $N = 2$  in Equation (6)) to “inverse”  $[(\exp(t\mathbf{D}))^2]_{(d,g)}$  because there are *two* summands ( $e^{(6+7)t}$  and  $e^{(5+4)t}$ ) in Equation (3). In the general case, we observe that the number of summands

for *arbitrary* element  $(u, v)$  of  $(\exp(t\mathbf{D}))^k$  in Equation (3) should be consistent with (i) the choice of  $N$  in Equation (6) and (ii) the number of paths with  $d$  hops between vertices  $u$  and  $v$  (that is,  $[\mathbf{A}^d]_{(u,v)}$ ).

*Example 2:* Consider the water network in Figure 6. To compute the sum of the pipe section lengths over all paths with  $d = 3$  hops between vertices  $b$  and  $i$ , we first obtain its distance matrix  $\mathbf{D}$  and adjacency matrix  $\mathbf{A}$  (see Example 1). Next, we evaluate  $[\mathbf{A}^3]_{(b,i)} = 3$  and

$$[(\exp(t\mathbf{D}))^3]_{(b,i)} = \underbrace{e^{(6+2+6)t}}_{b \rightarrow d \rightarrow e \rightarrow i} + \underbrace{e^{(6+4+4)t}}_{b \rightarrow d \rightarrow h \rightarrow i} + \underbrace{e^{(7+5+4)t}}_{b \rightarrow g \rightarrow h \rightarrow i}$$

Finally, choosing  $N = 3$  in Theorem 1, we can “inverse”  $[(\exp(t\mathbf{D}))^3]_{(b,i)}$  as follows:

$$\begin{aligned} & \lim_{t \rightarrow 0} \frac{3}{t} \log \left( \frac{1}{3} \left( [(\exp(t\mathbf{D}))^3]_{(b,i)} \right) \right) \\ &= \lim_{t \rightarrow 0} \frac{3}{t} \log \left( \frac{e^{(6+2+6)t} + e^{(6+4+4)t} + e^{(7+5+4)t}}{3} \right) \\ &= (6+2+6) + (6+4+4) + (7+5+4) = 44. \end{aligned}$$

Hence, the sum of the pipe section lengths over all paths with 3 hops between vertices  $b$  and  $i$  is 44. ■

After the nominator of Equation (1) is obtained, the “average length”  $[\mathbf{W}^{(d)}]_{(u,v)}$  follows immediately:

*Theorem 2:* The “average length” of all paths with  $d$  hops between every two vertices  $u$  and  $v$  can be computed as

$$[\mathbf{W}^{(d)}]_{(u,v)} = \begin{cases} \lim_{t \rightarrow 0} \frac{1}{t} \log \left( \frac{[(\exp(t\mathbf{D}))^d]_{(u,v)}}{[\mathbf{A}^d]_{(u,v)}} \right), & \text{if } [\mathbf{A}^d]_{(u,v)} \neq 0; \\ 0, & \text{if } [\mathbf{A}^d]_{(u,v)} = 0; \end{cases}$$

As a special case,  $\mathbf{W}^{(1)} = \mathbf{D}$ . This is because, when  $d = 1$  and  $u \neq v$ ,  $[\mathbf{A}^d]_{(u,v)} = 1$ . Then,

$$[\mathbf{W}^{(1)}]_{(u,v)} = \lim_{t \rightarrow 0} \frac{\log([\exp(t\mathbf{D}))_{(u,v)}]}{t} = \lim_{t \rightarrow 0} \frac{[(t\mathbf{D})]_{(u,v)}}{t} = \mathbf{D}_{(u,v)}.$$

*Example 3:* Recall Example 2. Since  $[\mathbf{A}^3]_{(b,i)} = 3$  and the sum of the pipe section lengths over all paths with  $d = 3$  hops between vertices  $b$  and  $i$  is 44, the “average distance” is

$$[\mathbf{W}^{(3)}]_{(b,i)} = 44/3.$$

Theorem 2 gives an effective way to evaluate the “average length”  $[\mathbf{W}^{(d)}]_{(u,v)}$  with a *fixed* number  $d$  of hops in terms of  $\mathbf{D}$  and  $\mathbf{A}$ . Based on  $[\mathbf{W}^{(d)}]_{(u,v)}$ , we can obtain the “average length” matrix  $\mathbf{S}^{(L)}$  within  $L$  hops as follows.

*Definition 2:* Let  $0 < \lambda < 1$  be a user-controlled decay factor. Given a water network  $G$ , its “average distance” matrix  $\mathbf{S}^{(L)}$  within  $L$  hops ( $L = 2, 3, \dots$ ) is defined by

$$[\mathbf{S}^{(L)}]_{(u,v)} = \begin{cases} \frac{1}{\beta} [\lambda \mathbf{D} + \lambda^2 \mathbf{W}^{(2)} + \dots + \lambda^L \mathbf{W}^{(L)}]_{(u,v)}, & (u \neq v); \\ 0, & (u = v). \end{cases} \quad (7)$$

where

$$\beta = \sum_{i=1}^L \lambda^i \cdot \mathbf{1}_{\{\mathbf{W}^{(i)}_{(u,v)} \neq 0\}}.$$

Here,  $\mathbf{1}_{\{\mathbf{W}^{(i)}_{(u,v)} \neq 0\}}$  is an indicator function, which returns 1 if  $\mathbf{W}^{(i)}_{(u,v)} \neq 0$ , and 0 otherwise. ■

Intuitively,  $[\mathbf{S}^{(L)}]_{u,v}$  captures the weighted average distance within  $L$  hops between vertices  $u$  and  $v$ . In Equation (7), the first term  $\lambda \mathbf{D}$  signifies that the paths of 1 hop have a contribution of  $\lambda$  to  $\mathbf{S}^{(L)}$ ; the second term  $\lambda^2 \mathbf{W}^{(2)}$  means that the paths of (longer) 2 hops have a (smaller) contribution of  $\lambda^2$  to  $\mathbf{S}^{(L)}$ , and so forth. The parameter  $\frac{1}{\beta}$  is a normalization factor, which guarantees that the sum of all the weighted factors  $\{\lambda, \lambda^2, \dots, \lambda^L\}$  in Equation (7) is 1.

The constant  $\lambda$  is between 0 and 1, which can be thought of as a confidence level. Empirically, it is set to 0.6–0.8, representing the rate of decay as the leak signature wave spreads across pipe sections.

*Example 4:* Recall the water network in Figure 6 and its distance matrix  $\mathbf{D}$  and adjacency matrix  $\mathbf{A}$  in Example 1. We choose  $\lambda = 0.85$  and  $L = 5$ . By Definition 2, the “average length” matrix  $\mathbf{S}^{(5)}$  can be obtained as follows:  $\mathbf{S}^{(5)} =$

	a	b	c	d	e	f	g	h	i	j
a	0	17.902	20.609	14.996	14.162	19.000	22.707	15.729	19.902	18.729
b	17.902	0	9.667	12.999	12.214	8.512	13.998	14.512	18.424	17.512
c	20.609	9.667	0	12.609	14.973	6.159	13.512	17.306	17.666	17.000
d	14.996	12.999	12.609	0	9.395	14.609	14.707	10.792	11.902	10.729
e	14.162	12.214	14.973	9.395	0	13.000	17.465	11.434	11.271	14.434
f	19.000	8.512	6.159	14.609	13.000	0	15.512	16.000	19.666	19.000
g	22.707	13.998	13.512	14.707	17.465	15.512	0	11.853	12.902	11.772
h	15.729	14.512	17.306	10.792	11.434	16.000	11.853	0	10.273	8.951
i	19.902	18.424	17.666	11.902	11.271	19.666	12.902	10.273	0	10.382
j	18.729	17.512	17.000	10.729	14.434	19.000	11.772	8.951	10.382	0

■

As opposed to the previous work [16] that considers only one single path of the shortest length,  $\mathbf{S}^{(L)}$  can capture multiple paths of different lengths between every two sensor locations by fully exploiting water network topology information. Thus, if the “average length”  $\mathbf{S}^{(L)}$  is used to quantify the wave distance travelled from a burst location to a sensor location, water loss events can be positioned more accurately, as will be shown in the next section.

### B. Effectively Localizing Water Loss Events

Having evaluated the “average length” matrix  $\mathbf{S}^{(L)}$ , we next present an efficient algorithm to position a water loss event with higher accuracy.

Our basic idea is to measure the difference in “average length” to two sensor locations that detect the burst transient at given times. Specifically, let  $\bar{v}$  denote the average wave speed, and let  $t_u$  and  $t_v$  be the time points when the burst transient event is detected at sensor locations  $u$  and  $v$ , respectively. Note that the sensor points in the water network are time synchronized, and the time of the burst event  $t_x$  is unknown in advance, but such a burst event must occur before  $\min\{t_u, t_v\}$  (earlier than either of the detected time



Figure 7: Water company research partner pipe rig.

at locations  $u$  and  $v$ ). We observe that the time gap between  $(t_u - t_x)$  and  $(t_v - t_x)$  (which can be calculated as  $|t_u - t_v|$ ) is mainly due to the difference in “average length” from the burst (source) location  $x$  to both sensor locations  $u$  and  $v$ . Hence, ideally we have the following equations:

$$\begin{aligned} t_u - t_v &= (t_u - t_x) - (t_v - t_x) \Rightarrow \\ \bar{v}(t_u - t_v) &= \underbrace{\bar{v}(t_u - t_x)}_{\text{dist}(u,x)} - \underbrace{\bar{v}(t_v - t_x)}_{\text{dist}(v,x)} \end{aligned}$$

which implies that

$$\begin{aligned} \bar{v}(t_u - t_v) &= \text{dist}(u, x) - \text{dist}(v, x) \\ &\approx [\mathbf{S}^{(L)}]_{u,x} - [\mathbf{S}^{(L)}]_{v,x} \end{aligned} \quad (8)$$

Then, we can enumerate each sensor location in  $V$  to find out the top- $k$  ( $k$  is often set to 3–5 in practice) that best approximate solutions  $\hat{X} \subseteq V$  of  $x$  to Equation (8), that is,

$$\hat{X} = \arg(\text{top-}k) \min_{x \in V} \{|\bar{v}(t_u - t_v) - ([\mathbf{S}^{(L)}]_{u,x} - [\mathbf{S}^{(L)}]_{v,x})|\} \quad (9)$$

Thus, the elements in  $\hat{X}$  form a “hyperbolic curve” with two focal points  $u$  and  $v$ . To determine the precise location along this “hyperbolic curve”, we need to choose another pair of sensor locations, say  $u$  and  $w$ , as two focal points, with the aim to produce the another “hyperbolic curve”, that is, to find out another set of the top- $k$  best approximate solutions  $\hat{Y} \subseteq V$  to the following equation:

$$\hat{Y} = \arg(\text{top-}k) \min_{y \in V} \{|\bar{v}(t_u - t_w) - ([\mathbf{S}^{(L)}]_{u,y} - [\mathbf{S}^{(L)}]_{w,y})|\} \quad (10)$$

The intersection of the two “hyperbolic curves”  $\hat{X} \cap \hat{Y}$  will produce a small number of possible locations where a water loss event may occur. Finally, we can search locally for the most likely water loss position along pipe sections connected to the closest sensor locations in  $\hat{X} \cap \hat{Y}$ .

## V. EXPERIMENTAL SETUP

The ultimate aim of this Smart Water project is to study and develop real-time control algorithms that take feedback from sensors and aim to respond to anomalies, demand changes, and reconfiguration in an optimal fashion.

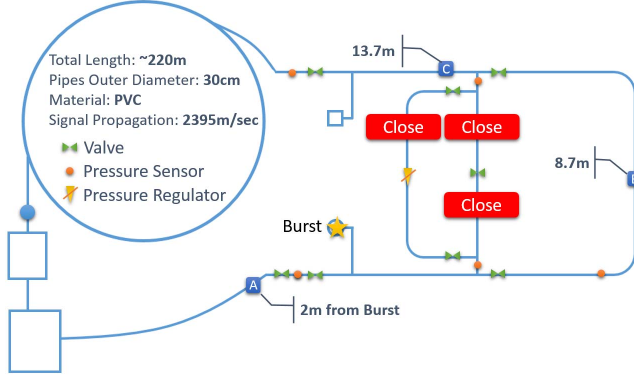


Figure 8: Pipe rig floor plan and infrastructure.

Given the security requirements of a live water distribution network, water companies typically have test rigs that are representative of the live network but which are more isolated and used for training staff etc. In this project we used such a test rig which was developed by our water company research partner<sup>1</sup> (Figure 7). The pipe rig was instrumented with sensor nodes based on Intel Edison development boards and NEC Tokin Ultrahigh-Sensitivity vibration sensors.

Figure 8 illustrates the infrastructure of the water network test rig. The rig has total length of pipes (30cm outer diameter) of 220 meters and contains 8 valves, 5 pressure sensors (with a sample rate of 10 samples per second), one pressure regulator, and one valve to emulate bursts and leaks. On top of this infrastructure, 7 nodes equipped with vibration sensors were deployed to the right area of the rig. Figure 8 presents only three nodes which produce non redundant data and are located in 2m, 8.7m, and 13.7m distances from the burst location respectively.

## VI. EVALUATION

Using the aforementioned infrastructure, a variety of burst emulations were conducted with different pressure levels and observed using vibration data (1000 samples per second) from 7 sensor nodes. For sake of brevity, the non-redundant data from the three sensor nodes of Figure 9 were used. Figures 10 illustrate the results of the in-node anomaly detection algorithm that resides in nodes (A, B, and C) and using 16000 data points. For this we represent the water burst on a highly pressurised water network. Table II aggregates the performance of the in-node decision-making algorithm based on average compression, anomaly detection accuracy, and communication savings for transmitting compressed data or only timestamps.

### A. Compression Rates

Data fluctuation influences the performance of the compression algorithm and the position of the node in the

<sup>1</sup>Our water company research partner has a policy whereby they remain anonymous in research publications.

topology means that data fluctuation rates differ. In this case the closest node to the burst (Node A) performs the lowest average compression rate. However even with this high level of fluctuation, the compression algorithm achieves more than 28% average compression rate.

### B. Anomaly Detection

The accuracy of the algorithm for all the nodes is greater than 90% for the dataset. The remaining 10% error is due to specificity as the current version of the algorithm remains slightly conservative. For subsequent work we would adjust the processing to filter this data and identify outliers.

### C. Communication Savings

Table II presents two different communication types. In the first type, the node transmits the compressed data only when an anomaly is locally observed with a result that reduces communication by 79.26%; this is for the worst case of (Node A) compared to the base-line scenario of periodic transmission which is how such data is relayed today. On the other hand, the second communication type is where we only send timestamps to the localization algorithm which saves more than 99% communication.

### D. Burst Event Localization

To estimate the burst location, we use the proposed methods as per Equations (9) and (10) to localize the water burst event. The sound velocity in PVC is 2395 m/sec. By using the time stamp information from all sensor locations in Figure 9, we can compute the difference in arrival times for every pair of sensor locations. Figure 10 illustrates the anomaly periods from the three individual nodes, as defined in server side, and allows easier interpretation of anomaly arrival time differences. First, we use the time difference between node *A* and *B*, which will produce a “hyperbolic curve” with *A* and *B* being two focal points. Next, we use the time difference between node *A* and *C*, which will yield another “hyperbolic curve” with *A* and *C* being two focal points. The two “hyperbolic curves” will intersect at node  $\star$  within an error radius no more than 0.5m. Note that the error radius is 0.5m because any burst event that occurs along the pipe section  $\overline{ab}$  (whose length is 0.5m) will produce the same time difference information. More precisely, as depicted in Figure 11, let *x* and *y* be the locations of any two burst events occurring along the pipe section  $\overline{ab}$ . Then, the time differences from any burst location to both *A* and *B* are exactly the same. That is,

$$\begin{aligned}
 |t_{(\star \rightarrow A)} - t_{(\star \rightarrow B)}| &= |t_{(b \rightarrow A)} - t_{(b \rightarrow B)}| \\
 &= |(t_{(x \rightarrow b)} + t_{(b \rightarrow A)}) - (t_{(x \rightarrow b)} + t_{(b \rightarrow B)})| \\
 &= |(t_{(y \rightarrow b)} + t_{(b \rightarrow A)}) - (t_{(y \rightarrow b)} + t_{(b \rightarrow B)})|
 \end{aligned}$$

This tells us that, by using the time difference information, the burst event can be determined inside a circle whose center point at node  $\star$  with guaranteed  $\pm 0.5m$  accuracy.



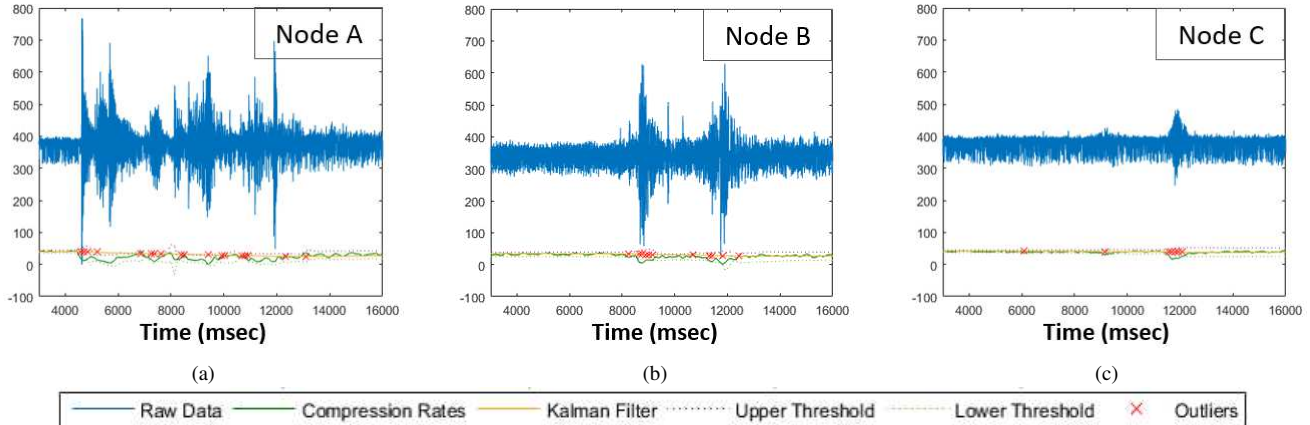


Figure 9: Anomaly detection from three sensor nodes (A, B, and C) which are located 2m, 8.7m, and 13.7m from the burst respectively.

Table II: Edge Processing Evaluation Results

	Average Compression Rate	Observed Anomalies	Real Anomalies	Anomaly Detection Accuracy	Communication Savings (Transmit Compressed Data)	Communication Savings (Transmit only Timestamps)
Node A	28.01%	18	20	90%	79.26%	99.88%
Node B	30.06%	11	12	91%	87.69%	99.93%
Node C	39.25%	8	8	100%	92.22%	99.95%

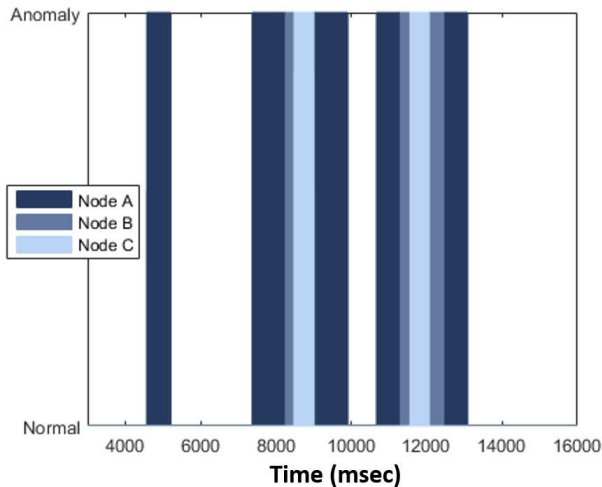


Figure 10: Anomaly detection periods.

## VII. PRIOR RELATED WORK

In the realm of Civil Engineering, there have been numerous mechanisms used to detect leaks using novel devices that mostly focus on intrusive sensing techniques such as in-pipe sensors or even mobile sensor bots that roam within the pipe network. Some of this prior work we touch on below, but a comprehensive review is too voluminous for this publication. Our work is instead focused on less intrusive approaches that can specifically form a cost-effective distributed control system to implement the next generation dynamic water

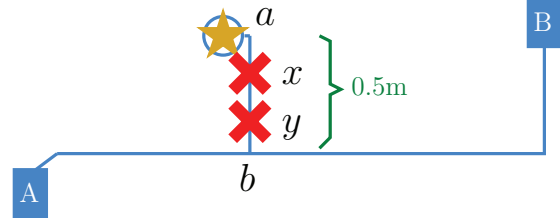


Figure 11:  $\pm 0.5\text{m}$  accuracy in localizing the burst event

network topology. The work related to this briefly discussed in this section.

**Edge processing.** In current smart water networks, physical states such as water pressure or pipe vibration are sampled at a reasonably high frequency (e.g. more than 128 samples per second) and stored in underground battery-driven sensor nodes. Currently, such raw data transmission is via cellular networks which is very costly. The alternative is to transmit over long-range (several kilometers) wireless communications, however high transmission power is required that leads to fast battery depletion. Current state of the art solutions [6] are unable to support underground battery driven sensor nodes and therefore the typical choice remains to be the expensive over ground sensor nodes directly connected to power and transmit data periodically every 15minutes using 3G.

Furthermore, these solutions use offline computationally

intensive algorithms to detect anomalies and require large amounts of historical data, leading to the inability of the algorithm to be distributed and deployable on sensor nodes. For example, [16] describes a wavelet-base algorithm which requires data continuously from the sensor nodes. To overcome these drawbacks, we have introduced an efficient stream-based edge data processing and decision making algorithm [9] which enables lightweight event-based communication.

**Water Burst Localization.** Over the last decade, there are some pioneering techniques proposed for water burst localization, such as wave propagation analysis [17], spectral clustering, and multiple hypotheses testing [18] (see [19] for a survey). Nonetheless, only a paucity of methods have been proposed in the context of a water network structure that explores graph topology. One excellent piece of work is Misiunas *et al.* [20] who leveraged a search-based technique to localize a burst point. Its main idea consists of two phases: in the first phase, the search is performed globally over all nodes in the network; in the second phase, if the burst is inferred to have occurred along the pipe, extra nodes are placed along each of the pipes and the global search is repeated. However, both steps of this method require that a global search over all sensor locations is performed. Therefore, its computation is expensive and will have scale issues in a water network with a high density of nodes.

Recently, Srirangarajan *et al.* [16] utilizes wave-based multiscale analysis of pressure signals to detect burst transients. To localize burst events, they also exploited the Dijkstra's algorithm for calculating the shortest distance between every two sensor locations. Nevertheless, we observe that, when a burst occurs, its wave may travel in all the possible directions of the paths (rather than down only one path in one direction with the shortest distance) from the burst location to sensor points. Thus, to accurately position burst events, it is not appropriate to depend only on the shortest travel time between two sensor locations.

We have some prior work on localisation using graph theory [21], [22], [23] but with pressure data only. Again, not only have we reapplied this work to use vibration data, which has some subtle differences, but we have integrated the end-to-end localization systems for the first time and successfully. That is, the edge anomaly detection and localization elements of the systems which produce a timely localization result and reduce the communication by 99% compare to the tradition periodic communication. Further, this is the first use of the combination of manual and automatic localization feedback information to tune the end-to-end system parameters to maximize its performance in a low-cost way.

### VIII. CONCLUSIONS

This paper presents a burst detection and localization scheme that combines lightweight compression and anomaly

detection with graph topology analytics for water distribution networks. We show that our approach not only significantly reduces the amount of communications between sensor devices and the back end servers, but also can effectively localize water burst events by using the difference in the arrival times of the vibration variations detected at sensor locations. Our results can save up to 90% communications compared with traditional periodical reporting situations. Further our localisation can find the position of the anomaly for our particular scenario within 0.5m error. This data driven approach is significantly better than many hydraulic modelling approaches that at best identify a leak to the length of a given pipe which can be 10s of metres [24].

In this paper we show that early transient or event detection can run on low resourced sensor nodes meaning that local control functions can occur with minimal latency and this paves the way for distributed control for next generation water networks. One can imagine that event time stamps are sent to the back end to be localized and this information fed into a control decision process (such as model driven control) where the pipe network would be reconfigured using remotely controlled valves to save both water and customer demand issues. Such networks ultimately mean a reduction of economic water scarcity due to poor water network operation and at the same time make use of scarce water resources.

### ACKNOWLEDGMENT

The algorithmic work presented here forms part of the Big Data Technology for Smart Water Nets research project funded by NEC Corporation, Japan. The sensing and water network data abstraction was part of the EU FP7 funded WISDOM: Water analytics and Intelligent Sensing for Demand Optimised Management project, FP7-ICT-2013-11 GA 619795. Many thanks to our Water Company research partner for allowing us time on their training pipe rig and assisting us to realistically emulate various pipe leaks under differing pipe conditions.

### REFERENCES

- [1] A. Johnson and J. Burton, "Water torture: 3,300,000,000 litres are lost every single day through leakage," *The Independent*, 2010.
- [2] I. Stoianov, L. Nachman, S. Madden, T. Tokmouline, and M. Csail, "Pipenet: A wireless sensor network for pipeline monitoring," in *Proc. IPSN*, 2007, pp. 264–273.
- [3] B. Aghaei, "Using wireless sensor network in water, electricity and gas industry," in *Proc. IEEE ICECT*, vol. 2, 2011, pp. 14–17.
- [4] A. Santos and M. Younis, "A sensor network for non-intrusive and efficient leak detection in long pipelines," in *Proc. IEEE WD*, 2011, pp. 1–6.

- [5] Z. Wang, X. Hao, and D. Wei, "Remote water quality monitoring system based on wsn and gprs," *Instrument Technique and Sensor*, vol. 1, p. 018, 2010.
- [6] M. Allen, A. Preis, M. Iqbal, and A. J. Whittle, "Water distribution system monitoring and decision support using a wireless sensor network," in *Proc. IEEE SNPD*, 2013, pp. 641–646.
- [7] "NEC and Imperial College London Smart Water Project," <https://wp.doc.ic.ac.uk/aese/project/smart-water/>, 2016, online; accessed 10-February-2016.
- [8] NEC/TOKIN, "NEC TOKIN sensors," [http://www.nec-tokin.com/english/product/piezodevice1/piezo\\_tech.html](http://www.nec-tokin.com/english/product/piezodevice1/piezo_tech.html), 2016, online; accessed 10-January-2016.
- [9] S. Kartakis and J. A. McCann, "Real-time edge analytics for cyber physical systems using compression rates." in *ICAC'14, USENIX*, 2014, pp. 153–159.
- [10] J. Kraus and V. Bubla, "Optimal methods for data storage in performance measuring and monitoring devices," in *Proceedings of Electronic Power Engineering Conference*, 2008.
- [11] Atmel, "Atmel AVR 8-bit and 32-bit Microcontrollers," <http://www.atmel.com/products/microcontrollers/avr/default.aspx>, 2015, [Online; accessed 9-February-2016].
- [12] I. M. Lab, "Filtering Sensor Data with a Kalman Filter," <http://interactive-matter.eu/blog/2009/12/18/filtering-sensor-data-with-a-kalman-filter/>, 2009, [Online; accessed 20-March-2014].
- [13] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *Proc. IEEE CDC*, 2005, pp. 8179–8184.
- [14] M. Mazo and P. Tabuada., "On event-triggered and self-triggered control over sensor/actuator networks," in *IEEE Conf. CDC 2008*, 2008, pp. 435–440.
- [15] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, pp. 55–66, 2010.
- [16] S. Srirangarajan, M. Allen, A. Preis, M. Iqbal, H. Lim, and A. J. Whittle, "Wavelet-based burst event detection and localization in water distribution systems." in *Journal of Signal Processing Systems*, 2013, pp. 1–16.
- [17] A. Radder, "On the parabolic equation method for water-wave propagation," *Journal of fluid mechanics*, vol. 95, no. 01, pp. 159–176, 1979.
- [18] J. P. Shaffer, "Multiple hypothesis testing," *Annual review of psychology*, vol. 46, no. 1, pp. 561–584, 1995.
- [19] G. Geiger, "State-of-the-art in leak detection and localization," *Oil Gas European Magazine*, vol. 32, no. 4, p. 193, 2006.
- [20] D. Misiunas, M. Lambert, A. Simpson, and G. Olsson, "Burst detection and location in water distribution networks," *Water Science and Technology: Water Supply*, vol. 5, no. 3-4, pp. 71–78, 2005.
- [21] W. Yu and J. A. McCann, "Efficient partial-pairs SimRank search for large networks," *PVLDB*, vol. 8, no. 5, pp. 569–580, 2015. [Online]. Available: <http://www.vldb.org/pvldb/vol8/p569-yu.pdf>
- [22] —, "High quality graph-based similarity search," in *SIGIR*, 2015, pp. 83–92. [Online]. Available: <http://doi.acm.org/10.1145/2766462.2767720>
- [23] —, "Co-Simrate: Quick retrieving all pairwise Co-Simrank scores," in *ACL*, 2015, pp. 327–333. [Online]. Available: <http://aclweb.org/anthology/P/P15/P15-2054.pdf>
- [24] A. Candelieri, D. Soldi, D. Conti, and F. Archetti, "Analytical leakages localization in water distribution networks through spectral clustering and support vector machines. the icewater approach," in *Procedia Engineering*, 2014, pp. 1080–1088.