

# Content-Centric Event-Insensitive Big Data Reduction in Internet of Things

Md Zakirul Alam Bhuiyan<sup>\*†</sup>, Guojun Wang<sup>†</sup>, Wang Tian<sup>‡</sup>, Md. Arafatur Rahman<sup>§</sup>, and Jie Wu<sup>¶</sup>

<sup>\*</sup>Department of Computer and Information Sciences, Fordham University, New York, NY 10458, USA

<sup>†</sup>School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China

<sup>‡</sup> College of Computer Science and Technology, Huaqiao University, Xiamen, China, 361021

<sup>§</sup>Faculty of Computer Systems and Software Engineering, University Malaysia, Pahang 26300, Malaysia

<sup>¶</sup> Center for Networked Computing, Temple University, Philadelphia, PA 19122, USA

**Abstract**—As more knowledge discovery functions or sensing units for event detection are added to sensor devices in the Internet of Things (IoT), devices acquire big data that is bigger than they are able to deliver using their radios in a given time window. As a result, energy consumption for big data acquisition and transmission and real-time data processing are great challenges. In this paper, we introduce BigReduce, a low-cost IoT framework for event detection that reduces a big amount of data at the time of data acquisition and before the data transmission across the network. BigReduce works on the analysis of the frequency content of signals as they are acquired and efficiently adapts the frequency rate based on the sensitivity to a respective event, such as fire event. Instead of transmitting the entire set of acquired data, BigReduce transmits only the signals that have a high event-sensitivity. We provide a detailed algorithm for fire event sensitivity indication based on the frequency contents. Results achieved through a lab testbed show that BigReduce is able to reduce energy consumption by at least 78% and data volume by 82% in comparison to other frameworks.

**Index Terms**—Internet of things, big data, data mining, decentralized signal processing, adaptive sampling, event monitoring, energy-efficiency

## I. INTRODUCTION

We are the witnesses to the development of emerging BigIo Data applications with bigger data volumes, veracities, and velocities [1]. Numerous sensing units integrated with a device are now found in the Internet of Things (IoT) for a lowwide variety of applications, including industrial control applications, mobile devices, automotive systems, structural health monitoring, healthcare, fire event monitoring, and climate monitoring. Sensor devices in the IoT are expected to function autonomously to support a long-lived and inexpensive acquisition of data. The technology that allows this to happen is decision-making or fusion, which leverages data collected from multiple sensing units and devices to get a more accurate and reliable view of the data than one would get by using the data from each discrete sensor device. When every device collects data at a high frequency rate, a big amount of data can create difficulties in decision-making in an event detection. However, they usually have inadequate resources to process them in the edge. In addition, various knowledge discovery functions for event detection are added to the sensor devices.

As a result, energy consumption for big data acquisition and transmission and real-time data processing are great challenges [2]–[4].

Those applications involve multi modality sensing, such as fire, damage, seismometers, audio, and imaging that produces raw big data. Frequent transmission of such raw data, even a reduced amount of data, results in significant data loss. All of the applications bring challenges to the sensors resource circumstances [5], [6]. If a system must lose data and it becomes difficult to analyze data at the base station (BS) and get event information, it is of course better to discard non-event sensitive data. For instance, the users (such as a fire event controller) of a fire event detection might be not interested in collecting data in quiet periods if there is an absence of fire [7]. As the frequency rate and frequency content of IoT devices is basically the defining factor of the application flow rate, carefully reducing the sampling rate after the analysis of the frequency content can satisfy the goals without any significant computation, communication cost or monitoring efficiency.

To address these challenges we introduce BigReduce, an IoT framework for event detection that reduces a big amount of data at the time of data acquisition and before the data transmission across the network. BigReduce works on the analysis of the frequency content of signals as they are acquired and efficiently adapts the frequency rate based on the sensitivity to a respective event, such as a fire event. Instead of transmitting the entire set of acquired data, BigReduce transmits only the signals that have a high event sensitivity.

Based on the analysis, each IoT device takes “short” and recurrent “bursts” of high frequency sampling. Depending on the analysis of the frequency content of the signals, if the signals collected in that sampling are sensitive to an event, the devices continue acquiring signals until acquired signals become insensitive to the event. Each device autonomously and automatically switches their sampling frequencies. We use a decision-making process algorithm whether or not there is event information in the event-sensitive frequency content. A remarkable change in the content implies the presence of an event. If an event really happens, the device needs to determinate all the collected data. Otherwise, in the case of no event, the acquired data is dropped; only the networking-

related data are exchanged in the IoT sensor networks.

The contributions of this paper are four-fold:

- We introduce BigReduce, an IoT framework for event detection that attempts to reduce a large amount of data.
- To reduce the data at the acquisition, we present a frequency content-centric signal analysis process.
- Based on the frequency content analysis, we present an event indication algorithm to tell the presence of an event locally. The whole amount of acquired data can be dropped if there is no event.
- We demonstrate the effectiveness of BigReduce through a lab testbed of IoT system implementation by using TinyOS and Imote2 sensors. The results indicate that our system can reduce a large amount of data and reduce a significant energy consumption.

This paper is organized as follows. We provide the design of the BigReduce framework in Section II. We provide event insensitive content reduction at the acquisition in Section III. Section IV presents the event-insensitive data reduction process before transmission in case of a fire event. Evaluation through a lab testbed implementation is conducted in Section VI. Finally, Section VII concludes this paper.

## II. BIGREDUCE: IOT FRAMEWORK FOR BIG DATA REDUCTION

Assume that a set of IoT sensor devices is given to monitor events in an IoT application. The events of interest include fires, structural damages, and mobile events. We consider the fire event as a representative event monitoring application, which requires multi-modal sensing. It involves a number of sensing units working together to detect fire events and provide a lot of data. Each IoT sensor device may have a number of sensing units, including humidity, smoke, temperature,  $CO$ ,  $CO_2$ , etc. The device can communicate among them using given communication ranges. Every pair of devices within a minimum range is allowed to share and compare their decision with their neighbors. Any two devices are connected if their corresponding neighbor devices are in their range and can communicate directly. The device may support discrete power levels [8].

We illustrate BigReduce in Fig. 1. At a given time interval, a device begins a short burst of samplings at a high frequency and examines these acquired signals to analyze the frequency content. The sampling duration of each burst of sampling at a high frequency sampling is followed by a low frequency sampling.

Whenever a device begins sampling, it is at a high frequency rate. If the frequency content is detected to be sensitive to an event (i.e., the changes in this frequency content are large), the sampling duration can be longer. Thus, this frequency content is important for calculating the event indication (as shown in Fig. 1). This frequency rate is kept until an analysis is made and the frequency content becomes insensitive. Once the frequency content is insensitive, the sampling situation becomes relaxed. A device is automatically switched between low and high frequency samplings depending on the frequency

content, which is due to the the absence/presence of an event. Using this technique in event indication, an analysis of frequency content is preserved in each discrete interval and, subsequently, a better frequency rate is selected. For the event indication in Fig. 1, a decision making computation algorithm based on the frequency content is used to decide whether the acquired data should be transmitted or dropped. This results in a reduced data acquisition and energy consumption if there is an event.

BigReduce contains a data reduction control, which is executed by every device autonomously. As shown in Fig. 1, through the control, BigReduce minimizes the amount of data in two phases: at the time of acquisition and before transmission (described in Sections III and IV). When enough frequency contents are acquired, the control is executed to choose an appropriate frequency rate. All of the sets of frequency contents are buffered in the device memory for future usage until they become unimportant.

## III. EVENT-INSENSITIVE FREQUENCY CONTENT REDUCTION AT THE ACQUISITION

In this section, we discuss the process of the event-insensitive frequency content data reduction.

The process is carried out at the time of data acquisition. This needs a frequency rate adaptation. The adaptation includes a few steps. In step 1, a device begins acquiring signals at a high frequency rate, and buffers them into a database (see Fig. 1). Every device acquires data in a given time interval. An interval is comprised of two types of sub-intervals: a high frequency interval and a low frequency interval. In the high frequency interval, a device begins with a short investigative frequency rate; in the low frequency interval, the remainder of the time is used that begins when the frequency rate is adapted to a lower frequency rate, based on the frequency content analysis, which can be influenced by the presence of an event. Thus, the adapted rate for lower frequency interval is adopted based on the required frequency rate,  $R_m$ .

In step 2, a frequency content analysis is carried out on the data points acquired during a high frequency interval to calculate the highest frequency content. If the sensor device discovers that the frequency content is sensitive, as shown Fig. 1, then the minimum frequency rate is the high frequency rate, i.e.,  $R_m = R_h$ . It is determined as:

$$R_m = c \cdot F_h \quad (1)$$

where  $c \geq 2$  is a confidence factor chosen to satisfy Shannon's sampling theorem [9]. However, high frequency rates are normally not known before data acquisition, signal activities, or the absence/presence of an event. To know this, step 3 is carried out that is to help to get the current frequency rate. At this step, a decision is made to know whether or not the current frequency rate needs to be continued or adapted. If a high frequency rate is still sensitive due to the presence of a possible event, the device continues data acquisition at the current rate; otherwise, the device adapts the frequency rate to a lower frequency rate. After getting the lower frequency rate, the device continues the data acquisition at this frequency rate.

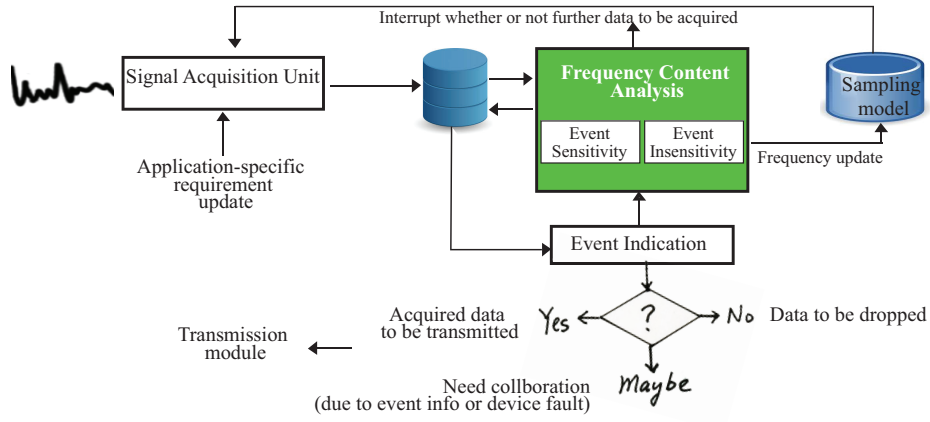


Fig. 1. Illustration of the BigReduce IoT framework.

#### IV. EVENT-INSENSITIVE DATA REDUCTION BEFORE TRANSMISSION IN CASE OF FIRE EVENT

Monitoring early residential fire events (e.g., in buildings, in aircraft engine compartments, etc.) or outdoor fire events (e.g., in forest) are critically important for extinguishing and reducing damages and fatalities. This is a high-data-rate application that acquires Big data. This requires a timely data acquisition, detection, and response. Unlike periodic sampling at a fixed-rate and other event detection applications, fire event detection is a narrow and specialized domain.

On the one hand, monitoring the absence/presence of a fire event involves composite event detection, where we require one or a combination of sensing units and a detection algorithm. The sensor device might be part of a network or work independently. On the other hand, a fire event is complex, composite (which is a combination of a set of atomic events), ambiguous, and vague in nature. In our fire detection algorithm, we assume that each sensor device is equipped with multiple sensing units, e.g., temperature, humidity, smoke, light, CO (carbon monoxide), etc. The use of more than one sensor device provides additional information on the environmental condition. Importantly, although an amount of the data is reduced by our adaptive sampling rate algorithm, a combination of all of the sensing units' input signals results in a huge amount of data transmission for the sink off-line detection compared to many other applications. We further reduce a large portion of data transmission through decentralized computing.

In the monitoring, we think that the *interesting data* is an answer to a question, which can be obtained by a set of predicates, not from the raw data in the form of numerical values. Moreover, users (or a monitoring operator) may not be interested in knowing the “exact” temperature or the smoke density of the monitored area. Instead, the users expect the quick answer to the concise question “is there any fire in the monitored area?”. We need to guarantee the accuracy and low-latency in the overall monitoring, where a conclusion indicates that a fire event should not be decided only based on one property of an atomic event. The algorithm simply shows how a fire event is defined [10].

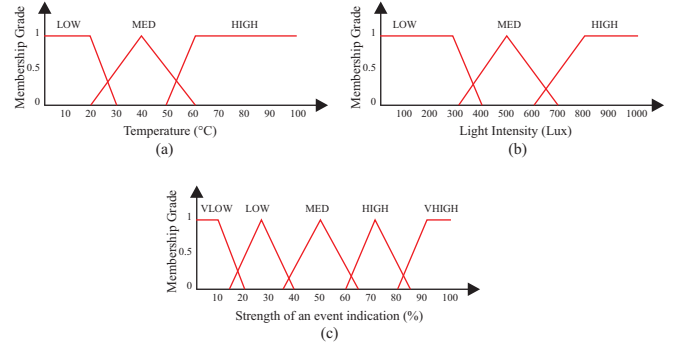


Fig. 2. Membership functions for the fire event detection: (a) temperature; (b) light intensity; (c) strength of an event indication.

#### Algorithm A. Fire Event Indication

```

1  DecentralizedFireEventComp{
2    on {
3      TempEvent e1 > HIGH AND
4      SmokeEvent e2 > HIGH
5    } where {
6      e2.time - e1.time = 10
7    }}

```

“DecentralizedFireEventComp” includes two atomic events. They must satisfy certain temporal constraints in order to indicate the presence of a fire event, which should satisfy some conditions, such as both the temperature and smoke density is HIGH, rather than a simple condition either temperature or smoke density are the HIGH (see the membership function for the fire event detection). A number of nodes with different sensing units, which are located in the area of the fire event, need to be involved and conduct local operations to give out the answer. To save energy, only the answer (i.e., the strength of event indication) is sent to the sink.

We define the following terms that are used in our fire event detection algorithm. *Sensed values* are the extracted captured frequencies from the frequency content. Each sensed value either belongs to a specific attribute or not. An *attribute*

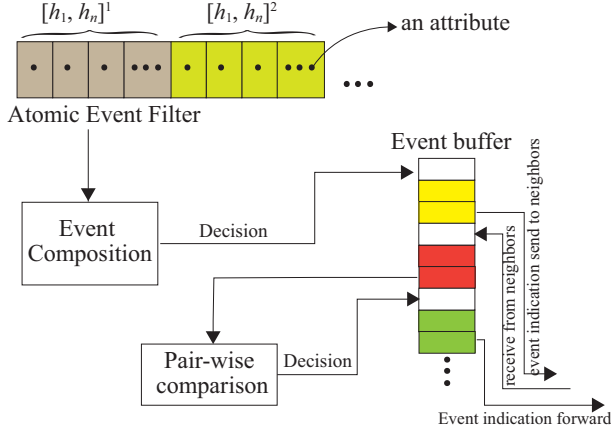


Fig. 3. Decentralized fire event detection framework.

is a collection of sensed values with some properties that correspond to a specific atomic event. That means each attribute contains an atomic event’s information. A *fire event* is a fusion or composition of multiple membership functions of multiple different types of attributes. A *membership function* is a function that is used to associate a grade to each collected sensed value, which has a weight between 0 and 1 over an interval of sensed values (see Fig. 2). Boolean operations are built up by composition of the membership grade of all types of attributes. Selection of the number of membership functions and their initial values are based on process knowledge.

**Definition [Membership function].** *This is a function that is used to associate a grade to each collected element, which has value between 0 and 1 over an interval of crisp variables. Selection of the number of membership functions and their initial values are based on process knowledge and intuition.*

The basic idea of the algorithm is as follows. The sensor device can produce a single data (say, a membership function) for each type of atomic event by simply aggregating a set of sensed values. Then, the sensor device uses some membership functions further in order to produce a small set of decision data for all of the atomic events. An event indication can be made by aggregating the decision data that contain greater information for the fire event than any one of the individual atomic event data alone.

Each sensor device may transmit an indication as a “fire emergency alarm.” The membership functions, VLOW (very low) LOW, MED, HIGH, and VHIGH are defined on the set of sensed values and on the attributes as shown in Fig. 2(a) and Fig. 2(b), and even on the strength of the event indication in Fig. 2 (c).

We use an event processing scheme as shown in Fig. 3 for fire event detection. In the scheme, atomic events are stored into buffers and associated with their types or attributes. Each atomic event attribute has a corresponding filter so that the sensor device can decide if the event occurs or not. The event filters consist of some predicates on the attributes. After data

collection in a certain  $D_h$ , a sensor device has frequency content. Thus, the range of captured frequencies,  $[h_1, h_n]$ , acquired at  $R_{h/a}$  is known. Consider  $[h_1, h_n]_1$  and  $[h_1, h_n]_2$  as the ranges of the frequencies of two atomic events of interest. Each sensor device computes the fire event indication by using the scheme in Fig. 3 that involves the following steps:

- Computes membership functions for each atomic event locally and then stores them in a separate table.
- Looks up the corresponding membership function for each attribute where the sensed values under the attribute have a graded membership in a real interval  $\{0,1\}$ .
- Evaluates the event against each membership function.
- Receives strengths of event indications from the neighboring nodes and transmits its own indications and does pairwise decision comparisons.
- If the fire event is evaluated to have happened, the sensor device triggers the transmitter and will transmit to the sink; otherwise, the sensor device keeps silent or sends an acknowledgment for reliability concerns.

Each node uses the following data structure to carry out above steps: a table for  $[h_1, h_n]_n$ , an event filter table (this table stores the membership grade for every attribute), a decision table (this table stores decisions and pairwise decisions), and an event forwarding table. Each sensor device can make decision based on all possible membership functions. For example, see some decisions on a fire event as follows:

- IF *Temperature* is HIGH and *Humidity* is HIGH and *Light-intensity* is LOW and *CO* is LOW, THEN the strength of the fire event indication is LOW.
- IF *Temperature* is MED and *Humidity* is LOW and *Light-intensity* is HIGH and *CO* is HIGH, THEN the strength of the fire event indication is high.
- IF *Temperature* is HIGH and *Humidity* is LOW and *Light-intensity* is HIGH and *CO* is HIGH, THEN the strength of the fire event indication is VHIGH.

After receiving all of the indications (or answers) sent by different sensors, accurate information about the presence of a fire event can be obtained at the sink by merging the indications. This detection technique is more suitable for a network platform because of resource constraints in the network; it is usually not efficient to send all the raw data to the sink for centralized event processing. Since the fire event is a composite event, it cannot be detected until all of the decisions about all of its atomic events are detected. If data about an atomic event is lost, the detection at the sink may not be accurate. We think that BigReduce can be very effective and efficient for “emergency alarming” applications compared to the methods of the static period and fixed-rate sampling in terms of accuracy, timely detection and energy cost.

## V. EVALUATION THROUGH A LAB TESTBED IMPLEMENTATION

### A. Methodology.

We use the same infrastructure, as shown in Fig. 13(a), and similar platforms, settings, and parameters as is used in SHM.

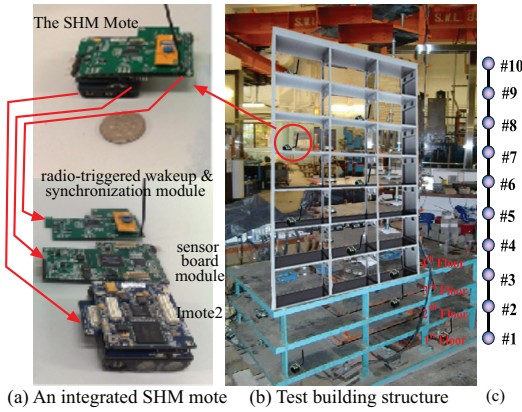


Fig. 4. (a) The SHM mote integrated by Imote2; (b) the twelve-story test structure and the placement of 10 SHM motes on it; (c) their deployment.

We implement a lab testbed system using TinyOS [11] on Imote2 sensor platforms [8]. We specially design a test infrastructure and deploy 10 Imote2 on it, as shown in Fig. 4(a). An additional Imote2 as the BS node is employed at 30 meters away. A PC is used as the command center and the command center is used for the BS and data visualization. We consider 4,096 data points to meet the fire event detection requirements. A successful ‘emergency’ event detection and response requires the presence of a physical event (such as a fire) accurately, which can be initially analyzed by calculating an event indication. Each Imote2 is interfaced to the main board via two connectors. The multifunction direct user interface provides a convenient and flexible solution to enable multiple sensor modalities. This interface provides a significant amount of built-in flexibility for different type of sensors. Besides 3-axes of acceleration used for fire event monitoring, we use the Imote2’s basic sensor board that is equipped with visible and IR light sensor (measurement range between 0.1 and 4000 lux), temperature sensor (measurement range between  $-40$  and  $123^{\circ}\text{C}@ + / - 2.5^{\circ}\text{C}$ ), and relative humidity sensor (measurement range between 0 to  $100\%RH@ + / - 5\%RH$ ). Note that CO or smoke detector sensors may be used for better detection performance in fire event monitoring. However, our current detection scheme is based on the three sensing units (i.e., temperature, humidity, and light) as we could not manage to integrate Imote2 with CO or smoke sensors.

For better observation, we conduct several experiments under different settings and compare the outcomes:

- **BigReduce-N**: This uses both data reduction methods but there is *no* event injection, i.e., we want to observe the performance of the network when it is assumed to be deployed in some applications where the presence of an event is rare;
- **BigReduce-C**: This is as is the same as used for SHM;
- **BigReduce**: We conduct three experiments over three different days (day1, day2 and day3) to distinguish the correctness of the monitoring.

### B. Physical Fire Event Injection.

We inject a fire event near the 5th sensor location. We place a lamp on the 5th floor that uses flammable hydrocarbon oil and emits heat around its location. We inject fire into the lamp exactly two hours after the network system runs. Then, we observe the sampling rate adjustment and event indication before and after the presence of the fire event. The sensor attached on 5th floor and its neighbors are expected to provide a HIGH temperature gradient, a LOW humidity gradient, and a HIGH light intensity value by estimating the membership function. That means that those nodes may provide HIGH strengths in the event indication. We consider both temporal and spatial gradients for this application. We increase the fire intensity on day3. Note that we omit discussing the reliability of the monitoring under sensor faults caused by the fire event.

### C. Experiment Results

Each sensor device acquires signals about nearby environmental information like temperature, humidity, and light intensity. As with the fire event injection, both the sampling rate and the interval vary based on the frequency content. The sample results achieved from the 10 sensor devices are shown in Table I. Table I shows the sample data that the sensor nodes collected. In *BigReduce*, each node filters the data from the frequency content. Based on the membership function, each device grades all the data and identifies the event as a percentage of the strength of the event indication and detects the event through our proposed detection scheme.

Looking into the details of the 5th sensor in Table I, we can see that the temperature is HIGH (81.1), the humidity is LOW (11), the light intensity is HIGH (775), the strength of the event indication is HIGH (75.3%), while the 3rd sensor device placed on the 3rd floor offers LOW (about 16.3%) and 6th sensor node offers HIGH. This is because a very lower temperature and humidity are detected by the 3rd sensor compared to others. In Fig. 2(a), a closer look reveals that the more intense the fire is, since the fire intensity is higher on day3 than day1 and day2. Users can make decisions on an ‘emergency alarm’ when the strength is LOW to MED (i.e., more than 15%). One can see that the strength is lower than 80% in *BigReduce*. We think that it would be higher if the fire intensity was VHIGH like severe real-world fire events in buildings and forests, and the design of *BigReduce* included smoke and CO sensors. Since the placement is on a bridge, which is not a square or a circle-shaped sensing field, the data traffic may not be evenly distributed by simply adding more sensor nodes.

We next analyze the amount of data reduction and networking at both phases and the networking, as shown in Table II. Different percentages of data reduction at both phases are given.

We next analyze the amount of data transmission reduction and the system lifetime. The results are shown in Fig. 5 (a). *BigReduce-N* and *BigReduce-C* show much difference in their lifetime. We can see *BigReduce-N* achieves a much higher lifetime under the absence of the event than

Table I  
Experiment Results Obtained from Different Sensors

Sensor no. #	Temperature (°C)	Humidity (%)	Light intensity (Lux)	Strength of the event detection indication (%)
3	30	52	450	16.3
4	55.5	34	505	40.7
5	81.1	11	775	65.2
6	88.5	8	729	75.3
7	73	17	620	64.4
10	55	25	300	25.2

Table II

Performance of Data Reduction in the IoT Networks

Data Reduction	Deployed sensor no. #									
	1	2	3	4	5	6	7	8	9	10
At the Acquisition	62.4%	47.1%	35.4%	32.1%	41.1%	55.7%	63.8%	82.1%	82.2%	92.1%
Before Transmission	63.8%	68.2%	42.5%	33.1%	37.3%	33.1%	65.3%	79%	93.2%	93.1%
Networking Data	(-) 3.9% (on average)									

BigReduce. This validates that event-sensitive sampling rate adaptations can be highly effective, especially in the applications where the presence of events is scarce. BigReduce enables net data reduction of 73.1% for the entire acquisition in the case of three sensing units of each sensor device, translating to a predicted 83.2% energy consumption reduction. The event indication identified through decentralized computing enables another net data reduction of 79.5% for the entire acquisition (in the second stage). Around 81.68% of the energy consumption is achieved by translating these data reductions, i.e., by the reduced transmission. The energy saved is due to decreased hardware activity, and translated to a reduction in the energy cost of the system from 0.68mAh to 0.04mAh in each round of monitoring. As a result, BigReduce can reduce the energy consumption around 78% in fire event monitoring.

Fig. 5(b) shows the system lifetime in different schemes. We analyze the lifetime under different settings. BigReduce-N shows has more lifetime than BigReduce where BigReduce achieves a better lifetime than others. Their poor performance is because their direct transmission to the sink and fixed interval and sampling rate selection, and bandwidth allocation. The outcome of the two real settings indicates that autonomous and dynamic adaptations of both the sampling rate and the interval are obvious to the network applications: 1) which are high-data-rate, data intensive, and emergency alarming, and/or; 2) which has a lower chance of the presence of events and a higher chance of data packet-loss.

## VI. CONCLUSION

We have proposed BigReduce, a novel IoT scheme for big data reduction for diverse multi-modalities sensing applications. Instead of transmitting hundreds of MB of data over the IoT network for off-line data analysis at the base station server, BigReduce focuses on reducing the energy consumption though big data reduction at the devices; hence

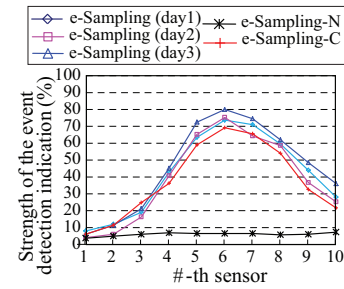


Fig. 5. Performance in different real settings: the strength of the fire event indication.

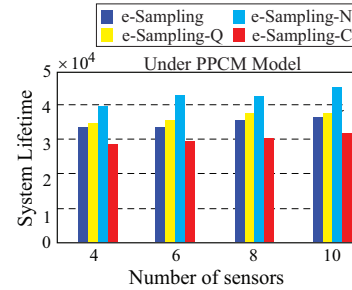


Fig. 6. Performance in different real settings: system lifetime.

reducing the lifetime of the entire IoT network system. A lab based testbed system is used to validate the benefits of BigReduce, which shows that, when both phases of data reduction are used, BigReduce reduce up to 78% of the energy consumed by Imote2 sensors.

## REFERENCES

- [1] M. Z. A. Bhuiyan and J. Wu, "Event detection through differential pattern mining in internet of things," in *Proc. of IEEE MASS*, 2016, pp. 1–8.
- [2] L. Xu, X. Hao, N. D. Lane, X. Liu, and T. Moscibroda, "Cost-aware compressive sensing for networked sensing systems," in *Proc. of IEEE IPSN*, 2015, pp. 130–141.
- [3] F. A. Aderohunmu, D. Brunelli, J. D. Deng, and M. K. Purvis, "A data acquisition protocol for a reactive wireless sensor network monitoring application," *Sensor*, vol. 15, no. 2015, pp. 10221–10254, 4 2015.
- [4] M. Z. A. Bhuiyan, G. Wang, and A. V. Vasilakos, "Local area prediction-based mobile target tracking in wireless sensor networks," *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 1968–1982, 2015.
- [5] Z. Chen, G. Barrenetxea, and M. Vetterli, "Share risk and energy: Sampling and communication strategies for multi-camera wireless monitoring," in *Proc. of INFOCOM*, 2012, pp. 1862 – 1870.
- [6] M. Z. A. Bhuiyan, G. Wang, J. Wu, J. Cao, X. Liu, and T. Wang, "Dependable structural health monitoring using wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–14, <http://dx.doi.org/10.1109/TDSC.2015.2469655>, 2015.
- [7] M. Z. A. Bhuiyan, J. Wu, G. Wang, T. Wang, and M. Hassan, "e-sampling: Event-sensitive autonomous adaptive sensing and low-cost monitoring in networked sensing systems," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 12, no. 1, pp. 1–29, 2017.
- [8] "Imote2 hardware reference manual," 2007, crossbow Technology Inc.
- [9] C. Alippi, G. Anastasi, M. Francesco, and M. Roveri, "An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 2, pp. 335–344, 2010.
- [10] S. Lai, J. Cao, and Y. Zheng, "Psware: A Publish / Subscribe middleware supporting composite event in wireless sensor network," in *Proc. of IEEE PerCom*, 2009, pp. 1–6.
- [11] TinyOS, <http://www.tinyos.net>.