IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 21, 2018, accepted December 11, 2018, date of publication December 19, 2018, date of current version January 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2886806

An Energy Efficient Integration Model for Sensor Cloud Systems

NGOC-THANH DINH AND YOUNGHAN KIM, (Member, IEEE)
Department of Electronics and Telecommunication, Soongsil University, Seoul 06978, South Korea

Corresponding author: Younghan Kim (younghak@ssu.ac.kr)

This work was supported in part by Institute for Information & Communications Technology Promotion (IITP) grants funded by the Korea government (MSIT) (No. 2015-0-00575, Global SDN/NFV Open-Source Software Core Module/Function Development) and in part by MSIT, Korea, under the Information Technology Research Center (ITRC) Support Program supervised by IITP, under Grant IITP-2018-2017-0-01633.

ABSTRACT This paper proposes a novel information-centric prediction-based integration model for WSNs and the sensor cloud, which exploits the trade-off between the data accuracy requirement of applications, sensing data prediction quality, and the energy efficiency of sensors, to reduce the workloads and energy consumptions of resource-constrained sensors. The model decouples information producers (IPDs) (i.e., physical sensors) from information providers (IPVs), which are implemented as IPDs' virtual sensors in the sensor cloud, to enable IPVs to provide sensing services when IPDs sleep. In the model, we design an efficient interactive sensing data prediction scheme for IPDs and IPVs to predict and control the accuracy of the sensing data prediction of IPVs using internal temporal information correlation, community detection, and external information correlation among sensors. According to the data accuracy requirement of applications, the model controls: 1) the number of IPDs required to be active and 2) when an active IPD transmits sensing data to the sensor cloud, to maintain the quality of sensing data meeting the requirement. Through extensive experiments with data collected from the real-world IntelLab sensor deployment, we show that the model achieves significant improvements in terms of data transmission suppression ratio, energy efficiency, and response latency compared with the existing schemes.

INDEX TERMS Information centric wireless sensor networking, energy efficiency, sensor cloud, IoT cloud, interactive sensor data prediction, machine learning, sensor quality of information, SDN/NFV.

I. INTRODUCTION

Periodic sensing data collection is one of the popular applications in Wireless Sensor Networks (WSNs) as well as the Internet of Things (IoT). In many practical IoT scenarios (i.e., pervasive computing applications, building management,. . .), applications or users are interested in periodically receiving sensing information with a given accuracy level. Therefore, the sensing data collection service is allowed within a given level of error tolerance [2]. For that reason, the data prediction and machine learning can be applied in the sensing data collection to improve the energy efficiency [2]–[4]. In current machine learning and sensing data prediction schemes [2]–[4] for WSNs, data packet transmission can be suppressed as long as the locally sensed data are compatible with the prediction model.

However, the current machine learning and sensing data prediction solutions are normally implemented inside WSNs (i.e., at a cluster head or normal sensor nodes), thus the solutions have a limited prediction capability as well as limited improvement for energy efficiency. The reason is that WSN nodes have a limited computing and storage capability (i.e., a few samples may be stored and processed for training predictors), as well as the constrained energy budget. For resource constrained sensors, only simple and small-scaled prediction scheme can be implemented. Additionally, the above prediction schemes only consider the local correlation [2], [4], [5] in sensing data (i.e., sensing data of sensors within a cluster) [2], [4]. As a result, a sensor has a lower chance to have good data sources to train the predictor. Globally correlation discovery is inefficient or even infeasible for resource-constrained sensors. Due to the resource-constrained nodes, many existing studies assume a priori correlation of sensing data among sensors (i.e., nodes within a cluster are correlated) [6], [7]. However, the correlation among sensors is various and may change over time. In many cases, two nearby nodes may have poor correlation while two sensors that are not geographically collocated may show a strong correlation. Moreover, although existing solutions enable sensors to suppress a number of packet transmissions, sensor nodes are still required wake up frequently for checking the prediction quality and making a decision to transmit their packets or not [2]. Therefore, the energy efficiency improvement is still limited. In addition, the existing solutions are not designed to support the sensor-cloud [8]–[11] which provides the sensing services to multiple applications having different requirements at the same time. Detailed literature reviews of this topic can be found in [2]–[4] and [12].

Regarding the recent evolution from the traditional WSNs to the sensor cloud architecture for the sensing service provision [8]–[11], this paper introduces a new perspective of information-centric approach for WSNs and cloud integration. The proposed integration model focuses on sensing information itself for sensing services, regardless how information objects are produced (i.e., physical sensors' sampling or prediction) and where they are produced (at physical sensors or in the cloud), as long as its quality satisfies the requirements. In particular, we propose an efficient information producer (IPD) (i.e., physical sensor) and information provider (IPV) [13]–[15] decoupling model for a semantic sensor-cloud integration. The concepts of IPD and IPV are borrowed from the information centric networking (ICN) [13], [14], [16], [17] although the perspectives may not be the same. While IPDs obtain their sensing data from the physical world, their IPVs that are implemented as the IPDs' virtual sensors on the sensor-cloud, are responsible for 1) providing the IPDs' sensing services to applications or end users, and 2) optimizing the workload for the physical WSN. The IPVs then provide the sensing services with sensing information that may or may not be generated by the IPDs as long as the data accuracy satisfies the requirement. By decoupling, IPVs store sensing data and make them available for applications as well as end users while most of IPDs can be put into the sleep state for energy saving.

The integration model is proposed to exploit the trade-off between energy efficiency of sensor nodes and the accuracy of their sensing data (i.e., obtained through machine learning or sensing data prediction techniques), with a respect to data accuracy requirements of applications. The proposed model takes data accuracy requirements of applications as the input parameters for building the prediction model and for optimizing the number of physical sensors required to be active to meet the requirements. Based on requirements of applications, IPVs are globally grouped into information correlated communities (ICC) (i.e., the same ICC is applied for corresponding IPDs) using the community detection theory and Least Absolute Shrinkage and Selection Operator [18], [19], without any prior assumption about the correlation. A prediction scheme based on the external information correlation is designed on top of the ICCs using the cross-correlation regression. We then establish a hybrid prediction model as a combination of both the external information correlation and the internal information correlation (sensing data correlation of a sensor in the time dimension) to enable IPVs to predict their IPDs' sensing data accurately as well as controllably without demanding the IPDs to wake up frequently. The proposed integration model demands only one IPD within an ICC to be active at a time for maintaining the prediction quality so that the model can schedule most of IPDs to sleep deeply. The active IPD plays the role as a prediction quality controller of its ICC and provides data sources for its ICC. In the proposed model, most of the complicated operations and computing tasks are offloaded for processing in the sensor cloud, so the model reduces the workload for constrained WSNs. Compared to the preliminary version [1], this paper provides more insight into the detailed design of the model such as adding the detailed application model, detailed operations for the ICC discovery and external information correlation-based prediction scheme. We design new mechanisms for the model including the adaptive predictors, the periodic validation, as well as mechanisms for scheduling and training phases. We extensively conduct new experiments and present significantly more results under various network conditions in comparison with the state-of-the-art schemes such as new results for the service availability, the percentage of required active sensors, the control overhead and the network lifetime.

In summary, this paper makes the following contributions.

- We propose a novel information-centric integration model for WSNs and the sensor cloud, which exploits the trade-off between the data accuracy requirement of applications and energy efficiency to reduce workloads and energy consumptions for resource-constrained sensors. The model decouples IPVs from IPDs (physical sensors), and enables IPVs (implemented as virtual sensors in the sensor cloud) to provide sensing services even when IPDs sleep, thus allowing most of the physical sensors sleep deeply.
- We design an efficient interactive prediction scheme for IPDs and IPVs to predict and control the sensing data prediction. In particular, the scheme is designed using an internal temporal information correlation, community detection, and external information correlation among sensors. According to the accuracy requirement of applications, the scheme groups IPVs and IPDs into highly information correlated communities. Only one IPD per an ICC is required to be active to update sensing data and control the accuracy of data predicted by the IPVs, while other IPDs are scheduled to sleep deeply for energy saving.
- The model can support almost any level of the data accuracy requirement of applications by controlling 1) the number of IPDs required to be active and 2) when an active IPD needs to transmit their sensing data to the sensor cloud based on the data accuracy requirement. For example, according to a higher accuracy requirement, small size ICCs may be discovered, so more IPDs are required to be active and they also need to transmit their sensing data to the sensor cloud more frequently.

Through extensive experiments and analysis with data collected from the real-world IntelLab sensor deployment [20], we show that the model achieves significant improvements in terms of data transmission suppression ratio, energy efficiency, and response latency compared to the existing sensor cloud model [8] and the state-of-the-art prediction schemes for WSNs [2], [4]. In addition, the model supports multiple applications at the same time and allows most of IPDs to sleep deeply with a long interval, not just suppressing their data transmissions like other schemes.

II. RELATED WORK

Recently, studies have shown drawbacks of traditional WSN models [8]–[11], [21], [22] like sensor management and sensing service models. Sensor-cloud [8]–[11], [21] is then introduced as a promising model. The sensor-cloud model exploits the high capacity storage, powerful processing and service distribution of cloud computing for sensing information. The cloud and WSNs integration enable providers to provide sensing-as-a-service in which sensing information collected from a WSN is distributed to multiple applications or multiple users at the same time, instead of the conventional sensing service model for a single dedicated user and application. In the sensor cloud model, the sensor cloud acts as a middleware between the cyber world and physical WSNs.

Some works have been carried out for sensor cloud designs using different ways for different applications. In the previous work, we propose and implement a sensor-cloud model for smart cities [9]. In another work, we design a location-centric sensor-cloud architecture for mobile cloud computing applications [21]. In [23] and [24], we solve the problem of end-to-end packet delivery latency control from physical sensors to the cloud. In a recent work [24], [25], we apply the sensor-cloud model to implement distributed interactive digital signage systems in which physical sensors are virtualized as virtual sensors implemented on the sensor-cloud using network function virtualization (NFV) [26], [27]. NFV is a new technology which enables providers and operators to implement network functions (NFs) as software, known as virtual network functions (VNFs), on standard servers using virtual machines (VMs), instead of using dedicated hardware. Our detailed implementation of the sensor-cloud and virtual sensors using NFV can be found in [25].

Fortino *et al.* [28], [29] redesign body sensor network (BSN) using the sensor cloud for body monitoring. Ghanavati *et al.* [30] and Hassanalieragh *et al.* [31] discuss challenges and applicability chances of the sensor cloud for e-health services. Misra *et al.* [32] propose to use virtual sensors for providing utility services for battlefield scenarios in the military. Neto *et al.* [33] implement an intelligent element integrated with industrial sensor cloud systems, and used in factory shop-floor to create digital machines based on sensing services. Lyu *et al.* [34] discuss scheduling issues for the sensor cloud used in smart living.

Most of the above works focus on solving specific problems in the applicability of the sensor-cloud while improving the cloud and WSNs is still an open issue. For resource-constrained physical sensor nodes, energy efficiency is one of the critical requirement. We find that in many WSN as well as sensor-cloud applications such as pervasive computing applications, building management,... applications or end-users are interested in periodically receiving sensing information with a given level of the data accuracy, thus sensing data collection is normally permitted to have an error tolerance bound [2]. For that reason, data prediction can be applied in the sensor-cloud model to reduce communication cost from sensors the sensor cloud [2]–[4].

In the literature, sensing data prediction and machine learning schemes for WSNs have been investigated [35]–[41]. In [35] and [36], sensors transmit their sensing data to their cluster heads which can generate new prediction models for each sensor. Cluster heads are also required to periodically update and transmit new prediction parameters to sensors. Jain *et al.* [37] uses spatial correlation to accesses the ability of cluster heads of computing prediction models for a sensor and select the best one for sensing measurement. Similar to [37], Min and Chung [38] incorporates spatial and temporal relations for prediction. In [39], the prediction is performed at sensors using GP regression. Each sensor node predicts the information that it will sample and adapt its scheduling. The purpose is to maximize information which it collects during a time interval. Raza *et al.* [40] use a naive mechanism for prediction, which is a type of linear approximation to compute the slope of sensing information's trend. Borgne *et al.* [41] show a study relying on pre-defined prediction model which results in poor prediction accuracy. In [2], OSSLMS is proposed as a dedicated hybrid model for data prediction, recovery, and compression to improve the efficiency of processing data at cluster heads. In [4], a hierarchical least mean square (HLMS) mechanism is used to speed up convergence rate and lower the mean squared errors in predicting data for WSNs. Extensive surveys of this topic are given in [2]–[4] and [12].

However, the existing sensing data prediction schemes are usually implemented inside WSNs (i.e., at cluster head or sensors), so they have a limited prediction capability and efficiency because sensor nodes are resource-constrained devices with low computing and storage resource (i.e., few sensing records can be stored and processed for the predictors' training) as well as limited energy capability. In addition, those prediction schemes only take into account the local correlation of sensing data (i.e., nodes within a cluster) [2], [4], [5], a node has a lower chance to have good data sources to train the predictor. For the sensor cloud, this paper is the first study which investigates an efficient model for WSNs and cloud integration by focusing on the sensing information itself, instead of physical nodes and exploiting external correlations among virtual sensors to increase the WSNs network lifetime.

The concepts of IPD and IPV are borrowed from the information centric networking [13], [14], [16], [17] although the perspectives are not the same. In this paper, we model sensors using an information centric approach [15] (ICWSN) which is studied in our previous work [13], [16], and [17]. In the ICWSN, entities (i.e., sensors or information objects) are named, instead of using IP address. As described in our previous works [13], [14], we use the name structure as follows. Each entity is named with a category prefix (CP) (i.e., "temp" for temperature sensors, "hum" for humidity sensors) and an unique ID, combining with an addressable and identifiable URI (Uniform Resource Identifiers) (i.e., bld A/fl1/roomx/ for building A/ floor 1/ room x/) to form a hierarchical name. In this way, the naming scheme is compatible with recent resource naming standardization released by ETSI M2M [42]. The benefit of the ICN paradigm is that data objects can be retrieved from any middleware and sensors to be easily discovered as well as grouped based on naming. We exploit this characteristic of ICN in implementation to facilitate the ICC discovery although the information centric networking [13], [15] is not required for the proposed model to work. The model can be implemented on any kind of networks.

III. THE ENERGY EFFICIENT INTEGRATION MODEL FOR SENSOR-CLOUD

In this section, we describe the design of the proposed energy-efficient sensor-cloud integration model and explain how we exploit IPVs (i.e., in the sensor-cloud) to improve the energy saving for IPDs (i.e., in physical WSNs) with a respect to the requirements of applications. Table 1 presents the list of acronyms used in this paper.

TABLE 1. List of acronyms.

Acronym	meaning
IPD	information producer
IPV	information provider
aIPV	the IPV of an active IPD
iIPV	the IPV of an inactive IPD
ICC	information correlated communities
NFV	network function virtualization
VNF	virtual network function
TDA	tolerance of data accuracy
SLA	service level agreement
ICN	information centric networking
STC	sensor type community
IIC	internal information correlation
EIC	external information correlation
CPM	closest-fit-pattern matching
CTP	collection tree protocol
LPL	low-power listening protocol
CCA	Clear Channel Assessment
CoAP	Constrained Application Protocol
TOSSIM	TOSSIM simulator in TinyOS
CC2420	a RF transceiver type for sensors

A. APPLICATION MODEL

The proposed model is designed with a focus on periodic sensing services of resource-constrained sensor nodes [42] in the sensor-cloud architecture [8], [11] where a WSN is normally deployed to provide sensing services for multiple applications at the same time. Our model enables multiple applications to request for periodic sensing services with their requirement (i.e., the sensing data accuracy level, in other words, the quality of information (QoI)). Based on the model, the sensor-cloud and the WSN interact in an efficient way for providing sensing data satisfying the requirements of the applications. We reuse the application model implemented in our previous works [8], [23], [24]. In the previous works, a downstream aggregation framework (i.e., application requests from the sensor-cloud to WSNs) for application requirements (i.e., sensing interval, latency requirement, or data accuracy requirement,...) has been developed for the sensor cloud. Assume we have N applications which request for sensing services provided by a WSN. The applications may have different requirements of the Tolerance of Data Accuracy (TDA) level. For example, some applications request for a TDA level of 5% (95% accuracy) while other applications may accept a TDA level of 10%. Note that the price of sensing services is usually proportional to the data quality requirement of applications based on a service level agreement (SLA) [43], [44]. For example, the higher the data accuracy level is requested, the higher the price is charged. The framework performs aggregation for all application requests to obtain the optimal consolidated TDA value TDA_c^{apps} (i.e., the minimum value of TDAs among TDA requirements of applications) which satisfies all application requirements. The purpose is to minimize the number of application requests which are sent to physical WSNs. Once the model finds a new consolidated requirement, related virtual sensors and physical sensors are adapted accordingly to meet the new requirement. The detailed operations of the application model and aggregation scheme can be found in [26] and [27], so we don't repeat in this paper.

In this work, we design an upstream enhancement framework for the sensing data collection from WSNs to the sensor-cloud.

B. THE DECOUPLING MODEL FOR IPD AND IPV

In the current WSNs [8], [13], [16], [17], requests of information consumers (i.e., applications and users) are normally forwarded to sensor nodes (i.e., IPDs), so a sensor node also plays a role of an IPV for processing requests as well as responses. This is obviously inefficient in term of the network lifetime because sensors are resource-constrained devices. Moreover, since resource constrained IPDs usually run in low power and lossy environments (i.e., sleep/awake mode), the availability of their sensing services is not always high. Requests and responses in such a constrained environment may experience high latency when a high traffic volume is processed inside WSNs.

We propose a decoupling model for IPDs and IPVs for reducing the workload for resource-constrained sensors and for enhancing the availability of sensing services.

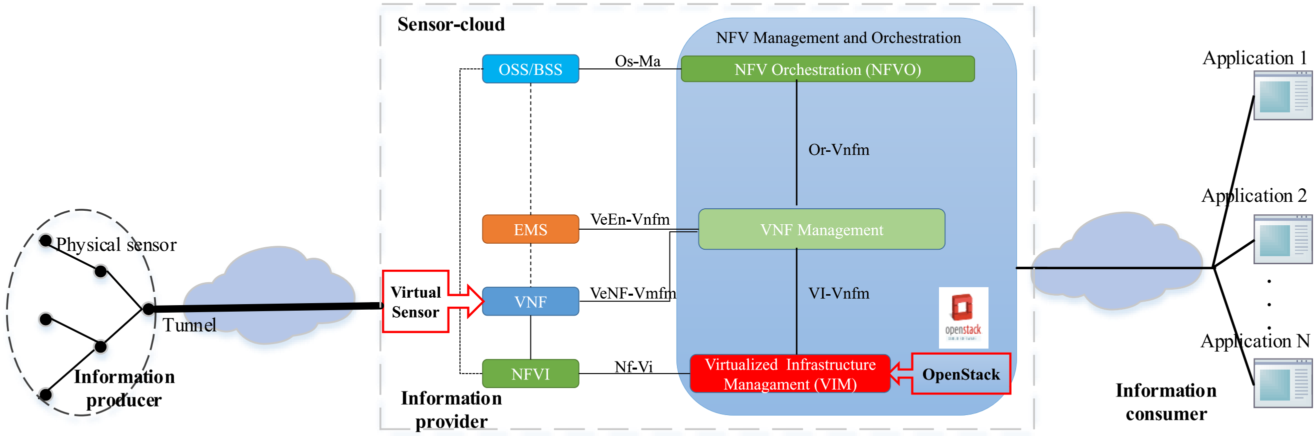

FIGURE 1. NFV-based information provider and information producer decoupling architecture for information centric sensor cloud.

In the proposed model, physical resource-constrained sensors operate as the IPDs while virtual sensors implemented in the sensor cloud play the role of IPVs to distribute the sensing data to other applications and users. As IPVs are implemented in rich-resource cloud environments, they are exploited to improve the energy efficiency of IPDs.

Although virtual sensors are also mentioned in the previous work [8], how to implement virtual sensors efficiently in the sensor cloud environment is still an open issue. In our point of view, if the sensor cloud is just simply a sensing data broker or middleware between applications and physical sensors, virtual sensors can be simply implemented as database objects. However, in our sensor cloud design [8], [9], IPVs are also explored to optimize performance and minimize resource consumption of physical sensors. Therefore, IPVs should have proper processing capabilities.

We implement IPVs as virtual network functions (VNFs) [26] in the cloud using Openstack [45]. As the NFV technology provides VNFs and the migration of network functions (NFs) from stand-alone hardware based on dedicated hardware to software appliances running on a cloud infrastructure, NFV can be used to deploy ICN solutions [46], [47] and virtual sensors easily. In the proposed model, IPVs inherit the properties from the IPDs and are considered as sensing data brokers, data caching points, and optimizers for physical sensors. IPVs can be implemented in the core clouds or edge clouds.

Depending on WSN scales, we can utilize one of the following two approaches to virtualize a WSN using NFV. In the first approach, each physical sensor (IPD) can be virtualized using one VNF. As a result, each IPV runs as a virtual machine deployed using OpenStack. In the second approach, the whole WSN can be virtualized within one VNF where the sink node is the head and each virtual sensor (IPV) is a VNF component (VNFC). Both approaches are equivalent. However, the latter is more suitable for small WSNs and highly constrained sensor nodes. This approach can be explored for virtualizing a greater scope of ICN objects in the Internet of Things (IoT). For example, the whole human body network can be virtualized for a semantic object named as "virtual me" to collect all information about a person. Similar concepts can be introduced such as a virtual home, virtual room, virtual factory, etc.

Figure 1 describes the NFV-based decoupling model for IPVs and IPDs referred to the ETSI NFV architecture [26], [27]. The detailed description of the general NFV architecture can be found in [26] and [27] and the detailed sensor-cloud implementation can be found in our previous works [25], [48], [49]. We briefly the architecture as follows. In the sensor cloud, the Virtualized Infrastructure Manager (VIM) manages computing, storage, network, and software resources (the implementation in OpenStack). VNF manager manages the VNF lifecycle management (i.e., instantiation, update, scaling, query, monitoring, fault diagnosis, healing, and recovery, etc.). NFV orchestrator automates the deployment and operations, as well as manages VNFs and VNFI. Element Management System (EMS) performs the typical management functionality for one or several VNFs. OSS/BSS is the operations support systems and business support system. Os-Ma, VeEn-Vnfm, VeNF-Vmfm, Nf-Vi, Or-Vnfm, VI-Vnfm are standardized interfaces defined in the NFV architecture. In the architecture, physical sensors in the sensor network are information producers. A secured tunnel is used for the communication between the WSNs and the sensor cloud. When the WSN and the sensor cloud are connected using an IP network, ICN packets at the sink are encapsulated with an IP packet header and then decapsulated at the sensor cloud. Multiple applications play the role of information consumers who may request for sensing services through the sensor cloud.

In the next part, we present how the proposed model exploits IPVs to enhance the energy efficiency of IPDs and improve the availability of sensing services based on our design of the interactive prediction mechanisms.

C. INFORMATION CORRELATION BASED SENSING DATA PREDICTION MECHANISMS FOR IPVS

Historical sensing data of a sensor may be correlated with the sensor's sensing information shortly. This is called the temporal correlation or internal information correlation (IIC). A number of prediction mechanisms for sensing data [2], [3], [50] have been studied in the literature as presented in section II. However, those mechanisms are normally used inside WSNs (at the sink or cluster head nodes) based on only the local correlation and IIC for the prediction. As a result, their improvement in the term of energy efficiency is limited as discussed in section I and section II.

We propose to extend the IIC based time series prediction (i.e., auto-correlation predictor) to the external information correlation (EIC) based time series prediction (i.e., using a cross-correlation predictor). We then implement a hybrid prediction model for IPVs in the sensor-cloud. The objective is to exploit IPVs to enhance the energy efficiency of IPDs by scheduling most of IPDs to go to sleep while their sensing services are still available and provided by the IPVs regarding to the applications' requirements. In our EIC based prediction scheme, we implement a cross-correlation predictor in each IPV. The cross-correlation predictor is designed to exploit the correlation in sensing data between an IPV and other IPVs. For that purpose, a mechanism is designed to enable each IPV to discover its highly correlated community so that the IPV can still have data sources to predict its IPD's sensing data and control its prediction accurately when its IPD is not unavailable (i.e., due to sleep or failure). In such as case, the IPV predicts its IPD's data based on 1) its historical data and 2) data of another highly correlated IPV whose IPD is active. All complicated and resource-consuming operations are processed by IPVs in the sensor-cloud to reduce the workload for physical sensors.

We classify two types of IPVs as follows. An IPV of an **active IPD** is denoted as **aIPV**. An IPV of an **inactive IPD** is denoted as **iIPV**. A different prediction mechanism is used for each type of IPV. The IIC based prediction is applied for aIPVs. A hybrid prediction model is used for iIPVs. In our model, active IPDs operate at a normal duty cycle mode with a wakeup interval T_w like in other schemes [2], [4], [22] while inactive IPDs sleeps deeply with a wakeup interval T_w^{deep} that can be much longer than T_w.

1) INTERNAL INFORMATION CORRELATION BASED PREDICTOR

The existing IIC based mechanisms for the predictor cannot be used directly for the application model as described above. The argument is that in our proposed model, applications and users are allowed to send requests with their selected requirement for the level of data accuracy. Therefore, the proposed model requires a controllable prediction accuracy. For that reason, although we implement the basic IIC-based time series prediction scheme [2], [3] for aIPVs, we add an adaptive transmission scheme for active IPDs and make the IPDs play the role of the prediction accuracy controller for IPVs.

In particular, for the IPV with its active IPD, both the IPV and its IPD run the IIC based autocorrelation predictor.

Auto-Correlation Predictor: For the auto-correlation predictor, an autocorrelation-based transversal filter is created as follows.

$$\omega(t) = \sum_{i=1}^{k} \omega(t-i)\delta(i) \tag{1}$$

where $\omega(t)$ is the predicted time series result at the time instance t; $\omega(t-i)$ with $i \in [1, k]$ represents the previous value of time series; $\delta(i)$ with $i \in [1, k]$ indicates k filter coefficients. The filter coefficients are determined based on the following linear equation which uses a set of T_k training data $[\omega(1), \omega(2), \ldots, \omega(T_k)]$ stored in a $(T_k - k) \times k$ matrix Ω available from the time series.

$$\begin{aligned}
\Omega\underline{\delta} &= \underline{\omega} \\
\Omega &= \begin{bmatrix} \omega(k) & \omega(k-1) & \ldots & \omega(1) \\ \omega(k+1) & \omega(k) & \ldots & \omega(2) \\ \vdots & \vdots & \ddots & \vdots \\ \omega(T_k-1) & \omega(T_k-2) & \ldots & \omega(T_k-k) \end{bmatrix} \\
\underline{\omega} &= [\omega(k+1), \omega(k+2), \ldots, \omega(T_k)]^T \\
\underline{\delta} &= [\delta(1), \delta(2), \ldots, \delta(k)]^T
\end{aligned} \tag{2}$$

The above linear equation is overdetermined because the number of equations $T_k - k$ is greater than the number of variables k. By using the least-square method [51], the filter coefficients are determined as follows.

$$\underline{\delta} = (\Omega^T \Omega)^{-1} \Omega^T \underline{\omega} \tag{3}$$

The filter coefficients are then used for predicting sensor data.

The auto-correlation predictor of an IPV is created in the sensor cloud at the end of the training phase as described in section III.D.4. After that, selected active IPVs transmit their filter coefficients using a request message to their corresponding active IPDs. The same predictor is then created at the active IPDs based on the received information. In this way, complicated computation is offloaded to the sensor cloud while the same predictor is used for both of the active IPV and its corresponding active IPD. The auto-correlation predictor of an active IPD and an active IPV can be adapted over time using a simple least-mean square filter technique if the prediction error does not meet the requirement of applications, as described in section III. F.

An active IPD controls the prediction accuracy of its corresponding IPV as follows. Upon each sensing measurement, the active IPD makes a comparison between its measured value and its predicted value. If the gap between the two values satisfies TDA_c^{apps} (i.e, the QoI requirement) of applications, the IPD suppresses its sensing data transmission for saving energy. If the gap between the two values does not satisfy the requirement, the IPD forwards and update its sensing measurement to its IPV. Using this simple control method, we enable IPVs to know whether their prediction is accurate enough or not. The control method also helps the IPDs reduce the number of packet transmission, thus saving more energy.

2) EXTERNAL INFORMATION CORRELATION BASED PREDICTION

In this section, we describe the establishment of a cross-correlation predictor which is created based on the EIC. For that, we first present how each IPV discovers its highly sensing information correlated community (ICC).

a: ICC DISCOVERY

We first define the concept of an ICC as follows. The ICC of an IPV node k (i.e., ICC_k) is defined as a set of other IPVs h that have highly correlated sensing data compared to k. Therefore, the sensing data of the IPVs in ICC_k can be exploited for predicting the sensing data of node k. The corresponding IPDs of the IPVs in ICC_k also form an ICC. This design is realized based on the fact that in practice, sensing data of sensors are normally correlated. However, how much the correlation between data vectors of IPVs is required to form an ICC depends on TDA_c^{apps} (i.e., the QoI requirement) of applications.

In the current design, the ICC discovery of an IPV node k is limited within its sensor type community (STC) for reducing the overhead and delay. An STC is defined as a group of sensors with the same type (i.e., temperature sensor) or can be extended as a group of sensors with sensor types which normally show a high correlation together (i.e., humidity sensing data and temperature sensing data normally show a high correlation). The sensor types are predefined. For simplicity, in this paper, we implement the ICC discovery using STC of the same sensor type only. The STC discovery is facilitated using the ICN naming scheme presented in our previous work [13] where sensors of the same sensor type (i.e., temperature) are assigned the same category prefix (i.e., "temp:*"). They are then grouped into the same STC, as illustrated in figure 2.

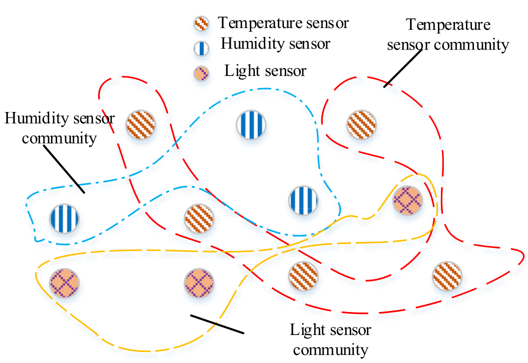

FIGURE 2. Information centric sensor communities.

Our ICC discovery is based on the network community discovery theory [19]. According to [19], we define constraints so that the prediction error for a data point of k (i.e., d_k) using the data from ICC_k is minimized.

Given a set of n IPV nodes $N = 1, 2, \ldots, n$, an iIPV node k and its data point d_k to be predicted, we find a community ICC_k and a regression function ω so that the expected error, expressed through the loss function $\Phi(d, \omega) = Loss(d_k, \omega(d_{STC_k}))$, is minimized.

We assume that ω is linear and Φ is mean square error (MSE). For accurate prediction, we find a set of nodes $ICC_k \in N \backslash k$ so that there is a decision ζ minimizing

$$\mathbb{E}[\Phi(D, \zeta)] = \mathbb{E}[(D_k - \zeta^T D_{ICC_k})^2] \tag{4}$$

In the equation, D_X is a random vector consisted of $\{D_k\}_{k \in X}$. Our ICC discovery is processed based on a historical sensor data set with m samples $D = [d_1 d_2 \ldots d_m]$. A heuristic solution can be implemented to find a decision ζ_k using the data set so that it can achieve the number of zero entries of ζ as great as possible. In other words, we should determine a sparse decision ζ_k to minimize the MSE and the L_1 norm, which is a typical Least Absolute Shrinkage and Selection Operator (LASSO) problem [18]. Therefore, the problem can be solved easily by introducing a LASSO parameter ρ as follows.

$$minimize\{\rho \parallel \zeta_k \parallel_1 + 1/2 \parallel d_{[-k]}\zeta_k - d_k \parallel_2^2\} \tag{5}$$

As a result, the ICC_k is determined as a group of IPVs with non-zero entries of ζ_k. The parameter ρ is used to controls the expected prediction error as well as the sparsity of ζ that affects the size of ICC_k.

A cross-correlation predictor is then established on the top of the discovered ICC. The predictor is implemented at IPVs to predict sensing data of iIPVs when their IPDs are not active.

b: CROSS-CORRELATION PREDICTOR

For the establishment of the cross-correlation predictor, we use a linear regression model with coefficients $[\gamma(0), \gamma(1)]$ [52] as follows.

$$\omega(t) = \gamma(0) + \psi(t)\gamma(1) \tag{6}$$

where $\omega(t)$ is the predicted time series result at time instance t; $\psi(t)$ is time series value of the active correlated node at time t. Similar to the auto-correlation, the coefficients here can be obtained through the following linear equation based on a set of the time series data Ψ of the active correlated node.

$$\Psi \underline{\gamma} = \underline{\omega} \tag{7}$$

With

$$\Psi = \begin{bmatrix} 1 & \psi(k+1) \\ 1 & \psi(k+2) \\ \vdots & \vdots \\ 1 & \psi(T_k) \end{bmatrix}$$

$$\underline{\gamma} = [\gamma(0), \gamma(1)]^T$$

3) A HYBRID PREDICTION MODEL FOR IIPV

We now obtain the hybrid prediction model for iIPVs. The hybrid prediction model determines the final predicted value for an iIPV by calculating the weighted average of the auto-correlation predicted value and the cross-correlation predicted value above with two weighted parameters η and θ as follows.

$$\overline{\omega(t)} = \frac{(\theta\omega(t)_{auto} + \eta\omega(t)_{cross}}{\theta + \eta} \tag{8}$$

where η and θ are set equal to the accuracy level of the cross-correlation and the autocorrelation, respectively.

4) TRAINING PHASE

To train the predictors and discover the ICC, we execute a training phase with the length $T_{training}$ during the network deployment or updated periods. During the training, all IPDs are set to be active and run a normal duty-cycled mode. To obtain data for training and ICC discovery, IPDs report their sensing information periodically to the sensor-cloud with a sensing interval I_s.

The size of an ICC is inversely proportional to the correlation requirement inferred by TDA_c^{apps} of applications, which is the input parameter of the LASSO solution above. The lower the TDA_c^{apps} requirement of applications the greater the size of an ICC can be discovered. To obtain highly accurate predicted values, the absolute requirement for the correlation coefficient of two nodes belonging to an ICC should be set close to 1. The value range of the correlation coefficient cc is $[-1, +1]$. The cc value of two nodes is 1 implying that the time series of sensing data of the two nodes show an identical trend. The cc value of two nodes is -1 implying that the time series of sensing data of the two nodes show an opposite pattern. If the cc value of two nodes is 0, sensing data of the two nodes show no correlation. At the end of this training phase, ICCs are discovered and the predictors are created.

The scheduling mechanism presented below shows that the proposed model needs only one corresponding IPD node per an ICC to be active at a time. The lower the correlation requirement results in the greater the number of nodes in an ICC. As a result, more nodes can sleep. However, this also leads to lower prediction accuracy. This trade-off between the energy efficiency and prediction accuracy is analyzed in the evaluation section.

D. SCHEDULING MECHANISM FOR IPDS

The proposed model requires to have only one aIPV per an ICC requires at a time (i.e., corresponding to one IPD to be active at a time) to maintain a proper accuracy level for sensing data prediction for all nodes meeting requirements of applications.

The aIPV and its active IPD are responsible 1) as a continuous data source which provides updated sensing records for the predictors and 2) as a prediction accuracy controller which updates the prediction model whenever the prediction error does not satisfy the given requirement. Other IPDs in the same ICC can sleep to save energy while their sensing data can be properly predicted by their corresponding IPVs.

In the proposed model, the scheduling works as follows. If an IPD is scheduled as an active node, its upper stream nodes (i.e., the parent nodes on its path toward the sink to the sensor-cloud) are set to be active too so that its sensing data can be forwarded to the sensor-cloud. In other words, the model maintains a connected graph for active IPDs toward the sink node.

Energy balancing is also considered in the proposed model to improve the network lifetime of sensors. For that purpose, the time frame is divided into rounds with a length RL. Note that the length RL is normally set long enough (i.e., 1 hour) and $RL \gg I_s$, the sensing interval of a sensor. In each round, the scheduling mechanism selects an IPD as the active IPD which has the highest residual energy for each ICC. Other nodes in the same ICC can sleep for saving energy. The scheduling is executed from the sink and repeatedly processed until the schedule of all sensors is set. In this way, the energy balancing among sensors is provisioned to improve the lifetime of the physical WSNs.

E. ADAPTIVE PREDICTORS AND PERIODIC VALIDATION

1) ADAPTIVE PREDICTORS

During the operating periods, the prediction model of an IPV is adapted on-demand to meet the requirement of the applications. In particular, when an active IPD detects its prediction error exceeding the given requirement (by comparing its predicted values and its measured values), the active IPD adapts its filter coefficients using the least-mean square filter technique to adjust its auto-correlation predictor. Firstly, the prediction error is calculated as follows.

$$\epsilon(t) = \omega(t)_{actual} - \omega(t)_{predicted} \tag{9}$$

The prediction error is then used to adapt the filter coefficients as follows.

$$\underline{\delta} = \underline{\delta} + \lambda\underline{\omega}\epsilon(t) \tag{10}$$

where $\underline{\omega} = [\omega(t-1), \omega(t-2), \ldots, \omega(t-k)]^T$ is the input vector of the filter, and λ is learning rate of the updating algorithm which is set following [53].

After adapting the auto-correlation predictor, the active IPD transmits the measured sensing data to its IPV. Upon receiving the measured sensing data, the IPV also updates its predictor in the same way. Therefore, an IPD and its corresponding IPV always use the same predictor. //

2) PERIODIC VALIDATION

As the information correlation among nodes is unknown prior and may change over time, revalidation is required. The proposed system performs validation periodically at the beginning of each round, to ensure that the predicted sensing data meet the requirement of applications. Note that the length of a round equals to N sensing intervals. In the validation period with length $T_{updating}$, all IPDs resume to their normal duty cycle mode and transmit their measured sensing data with a sensing interval I_s to the sensor cloud. IPVs obtain new measured sensing data to validate their prediction model. Correlation among nodes may be updated if changes are detected. After the validation completion, the scheduling mechanism is performed again to assign new active IPDs and inactive IPDs, for energy saving and energy balancing. In our reserved design, the validation can also be triggered once P consecutive predicted values don't meet the requirement of applications or a new consolidated TDA is found upon a new application request.

IV. PERFORMANCE EVALUATION

In this section, we present the performance evaluation of the proposed model with experimental and analysis results in comparison with state-of-the-art sensing data prediction schemes [2], [4] for WSNs and the existing on-demand sensor cloud design, O-SC [8] We conduct extensive simulations with sensor data collected from the real-world IntelLab sensor deployment [20] consisting of one sink node, 108 sensors totally with 54 temperature and 54 humidity Tmote-skype sensors. As implemented in our prior study [54], HTTP-CoAP converter [54] are reused in this paper for converting HTTP application requests to CoAP requests for sensor nodes. We assume that each application or user demands one of the sensing data types above. In the sensor-cloud, we encode application requests using XML templates and decode with SensorML interpreter for sensors [23], [24].

TABLE 2. Parameters.

Parameter	Value	Parameter	Value
TDA_c^{apps}	$2-20\%$	RL	$[50,100]*I_s$
channel sampling	$10ms$	$T_{training}$	$500*I_s$
$T_{updating}$	$[5,10]*I_s$	I_s	$31s$
T_w	$1s$	T_w^{deep}	$31s$
collection period	$3000*I_s$	$correlation$	$0.8-1$

For the WSNs, we use CTP and LPL [55], [56] as the sensing data collection protocol and the duty-cycled MAC mechanism. For the radio noise model, we use the closest-fit-pattern matching (CPM) [55]. We implement counters to track changes and record the time in each radio state of sensors to measure the duty cycle of a node. We set the CCA check parameter up to 400 times, as same as the default value used in the TinyOS LPL. The detailed configurations for simulations are presented in Table 2. Other parameters are kept the same as the default configurations of TOSSIM CC2420 radio model [55], [56]. The naming scheme [13] is used for sensors to facilitate the ICC grouping. In theory, for energy saving, the T_w^{deep} can be set equal to RL because deep sleeping sensors don't perform any sensing samples during a round. However, to support new application requests and validation triggering quickly during runtime, we set T_w^{deep} to 31 s only, equal to the sensing interval of sensors. The detailed implementation of a sensor-cloud prototype with Network Function Virtualization (NFV) [26] in OpenStack Newton version [45] is presented in our previous work [25], [49].

A. THE SENSOR CLOUD MODELS' COMPARISON

In this subsection, we compare the packet transmission overhead of the proposed model with the existing on-demand sensor-cloud design, O-SC [8]. For the experiments, we generate 6 sensing service requests from 6 different applications. Each request (i.e., from 1^{st} to 6^{th}) in the ascending order is sent to the sensor-cloud at a random time. We assume the *TDA* requirements of applications' requests are 20%, 15%, 10%, 5%, 2%, and 1%, respectively. We then obtain the number of packet transmissions in each model when the application requests are sent. In both of the model, the aggregation scheme [8] for application requests is performed. We present the obtained results in figure 3.

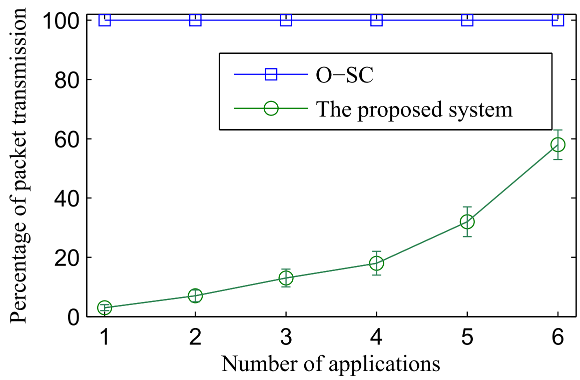

FIGURE 3. Percentage of packet transmission comparison.

The figure shows that the proposed model achieves a significant reduction in the number of packet transmissions of physical sensors, in comparison with O-SC. In particular, when only the first application with the *TDA* requirement of 20% requests for sensing services, the proposed model requires only three percents of packet transmissions in comparison with O-SC, for satisfying the same data accuracy requirement. The reasons are as follows. Firstly, with such a high *TDA* level, many sensors with an appropriate correlation can be grouped into an ICC. Therefore, more sensors within an ICC are scheduled to sleep, which do not transmit any packets. Secondly, active IPDs also transmit few packets as their prediction error is easily lower than the *TDA* requirement of 20%. In the case of O-SC, all sensors are set to be active and operate with the duty-cycled mode, and all sensors with packet transmissions transmit their sensing data to the sensor-cloud.

When requests with the lower *TDA* requirements (i.e., higher data accuracy levels) are sent, the proposed model requires a higher percentage of data packet transmissions from IPDs, to meet the requirement. For example, the proposed model requires as high as 60% of data packet transmissions to satisfy the *TDA* requirement of 1%, in comparison with O-SC. In such a case, only nodes with very highly correlated sensing data can be classified into an ICC and the

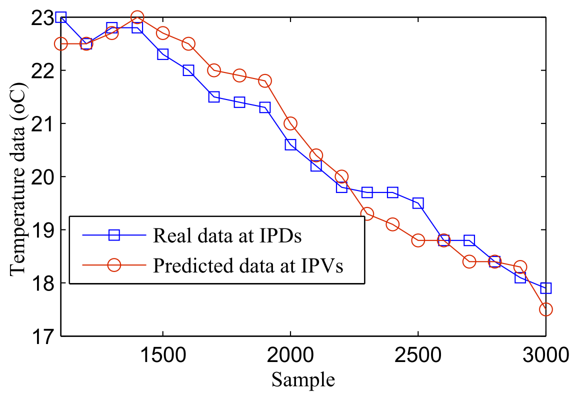

FIGURE 4. **Predicted values and real data comparison.**

packet transmission reduction mostly relies on the internal correlation.

The results indicate a clear trade-off between 1) the sensing data accuracy requirement of applications and 2) the energy efficiency enhanced for sensors. The proposed model exploits the above trade-off efficiently to improve the energy efficiency of sensors in WSNs when possible, regarding the *TDA* requirements of applications. Figure 4 presents a closer look into the correlation between the average measured sensing data and the average predicted sensing data of temperature sensors in the case of $TDA_c^{apps} = 5\%$. The figure shows that the predicted values always fluctuate around the measured values within a small gap to meet the requirement. In this way, the proposed model efficiently maintains the data accuracy satisfying the application requirements.

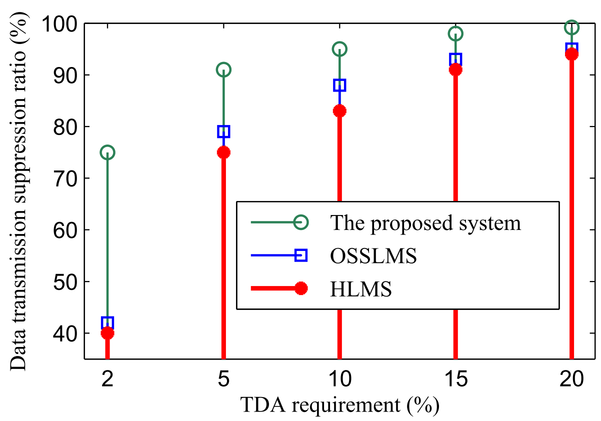

FIGURE 5. **Packet transmission suppression ratio comparison in a network consisting of one sink and 108 sensors.**

B. PACKET TRANSMISSION SUPPRESSION RATIO COMPARISON

Figure 5 shows a comparison in term of the average packet transmission suppression ratio of the proposed model with OSSLMS and HLMS under different TDA requirements. The figure shows that the suppression ratio for data packet transmissions of the proposed model is significantly higher than OSSLMS [2] and HLMS [4].

The results can be explained as follows. Firstly, in the proposed model, we implement the prediction scheme at IPVs in the sensor-cloud with rich resources. As a result, the predictors at IPVs can be trained using a larger dataset and more complex mechanisms can be implemented for the higher efficiency, compared to OSSLMS and HLMS as presented in section II. In the cases of the two local prediction schemes, OSSLMS and HLMS can store only a few sensing data records for training the predictors and simple operations can be applied due to the resource-constrained issues of physical sensors, thus limiting the prediction capability. For that reason, the higher the data accuracy requirement the greater the improvement ratio the proposed model can achieve compared to OSSLMS and HLMS. For example, the gap between the suppression ratio of the proposed model with that of OSSLMS and HLMS at the TDA requirement of 2% is greater than the gap at the TDA requirement of 5%, 10%, 15%, and 20%.

Secondly, the proposed model exploits the external correlation among nodes in the network for prediction. This enables a significant number of sensors to go to sleep, thus not incurring any transmission, while only one active node within an ICC is responsible to control the prediction quality. In OSSLMS and HLMS, only the correlation of nodes in local (i.e., within a cluster) is taken into account. In addition, each IPD is required to control the prediction quality itself, so all nodes are required to be active or in the duty-cycled mode. This does mean that all nodes may potentially generate packet transmissions.

C. THE SUPPRESSION RATIO COMPARISON IN LARGER SCALE NETWORKS

We now conduct simulations to evaluate the packet transmission suppression ratio of the three schemes in a larger scale and higher redundant WSN network. To have a larger scale network, we duplicate the current network of 108 sensors to have a network consisting of two sub-networks with 216 sensors totally. Each sub-network has the same number of 108 sensors and a sink node. The same data set is used. The purpose is to create a larger scale network and to create a higher redundancy of sensors across the two sub-networks. We carry out the same experiments and measurements as presented in the previous subsection. Figure 6 shows the packet transmission suppression ratio of the proposed system, OSSLMS, and HLMS. By comparing figure 5 and figure 6, we find that while the suppression ratio of OSSLMS and HLMS does not change, the proposed system achieves a significant higher suppression ratio in the new network. For example, the proposed system helps reduce 88 % of the number of data transmission, compared to only 75 % of that in the case of IV. B. The proposed system achieves this improvement because it considers the external correlation of all nodes in the network. As a result, the higher redundancy level of new sensor deployments across the networks is also exploited. This is one of the key advantages of the proposed system.

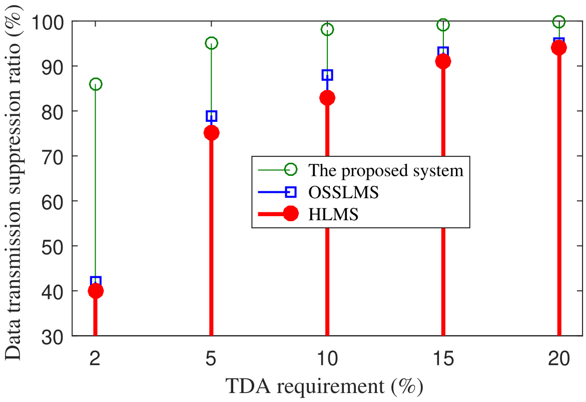

FIGURE 6. Packet transmission suppression ratio comparison in a network consisting of two sinks and 216 sensors.

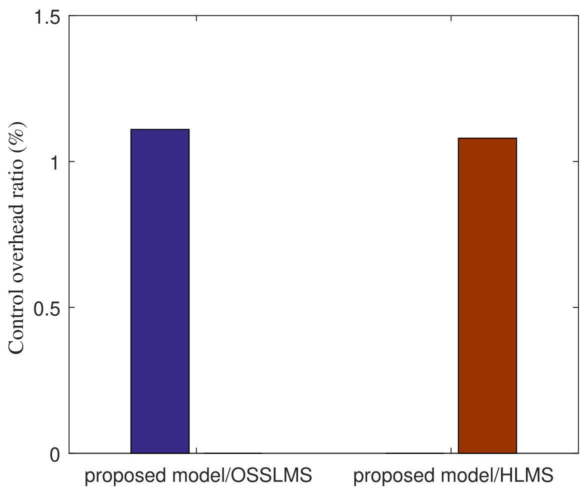

FIGURE 7. Control overhead ratio comparison between the proposed scheme with OSSLMS and with HLMS.

D. CONTROL OVERHEAD COMPARISON

Figure 7 shows the control overhead comparison between the proposed model with OSSLMS and with HLMS. The proposed model experiences a higher control overhead, between 7% and 11% higher compared to HMLS and OSSLMS, respectively. This is a trade-off of the proposed model to achieve a higher transmission compression ratio, as presented in figure 5 and figure 5, and a low percentage of required active sensors as shown in figure 8 and figure 9.

E. PERCENTAGE OF REQUIRED ACTIVE SENSORS

Figure 8 illustrates the percentage of required active sensors of the proposed scheme in comparison with HLMS and OSSLMS under various TDA requirements. In HLMS and OSSLMS, all of the resource-constrained sensors are required to be active in every wakeup interval. Although the two schemes allow data transmission suppressions, the sensors still have to wake up frequently every cycle to listen for transmissions and to perform sensing. The proposed model allows a high percentage of physical sensors to sleep deeply for energy saving. The percentage of required active sensors is inversely proportional to the TDA requirement.

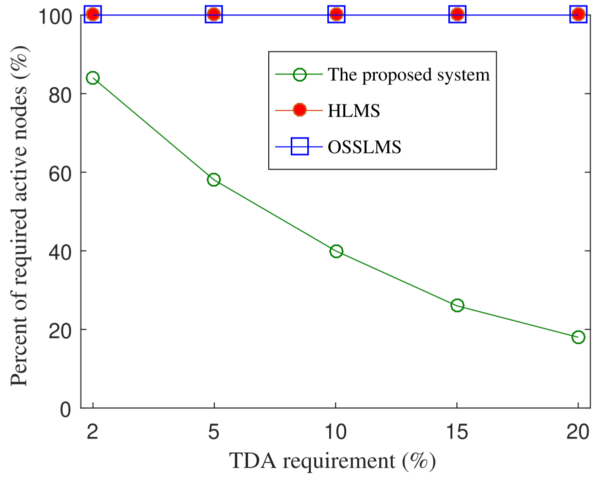

FIGURE 8. A comparison of the percentage of required active sensors in a network consisting of one sink and 108 sensors.

In particular, the lower the TDA requirement, the higher the number of physical sensors is required to be active for prediction control and for transmitting measured sensing data.

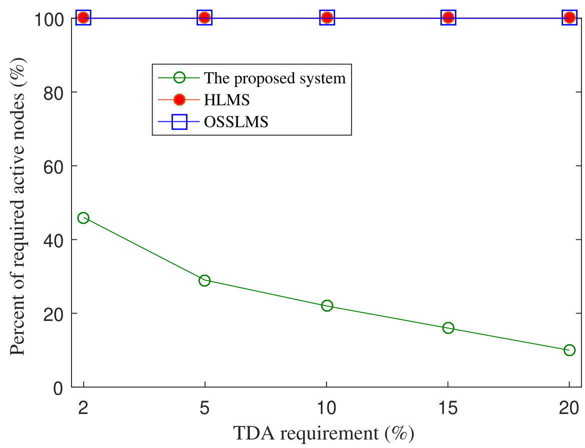

FIGURE 9. A comparison of the percentage of required active sensors in a network consisting of two sinks and 216 sensors.

Figure 9 presents similar results obtained with the network consisting of two sub-networks with 216 sensors described in IV. C. With a duplicated network to create a larger scale network and a higher redundancy of sensors across the two sub-networks, the obtained results of the proposed system are reduced significantly compared to those presented in figure 8. In particular, the percentage of required active sensors is reduced nearly a half. This means that more physical sensors are allowed to sleep deeply. The proposed model exploits the sensor redundancy well to allow most of the physical sensors to sleep deeply while HLMS and OSSLMS are not beneficial from the higher sensor redundancy across the network.

F. WSN LIFETIME COMPARISON

Figure 10 shows a comparison among the three schemes in term of the average duty cycle of IPDs. The figure indicates the overall energy efficient benefit achieved by the proposed model, compared to HLMS and OSSLMS.

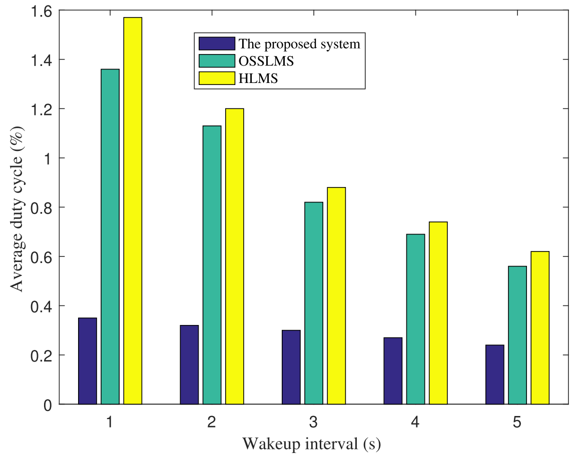

FIGURE 10. **Average duty cycle under various wakeup interval values.**

In particular, the proposed model witnesses significant lower average duty cycle results of IPDs in all wakeup interval settings. The results imply that the proposed model can achieve a longer network lifetime compared to HLMS and OSSLMS. The results can be explained as follows. Firstly, the proposed model requires fewer packet transmissions to meet the same *TDA* requirement. Secondly, the proposed model allows most of physical sensors sleeping deeply in a long period while HLMS and OSSLMS require all IPDs to wakeup periodically every duty cycle for checking their prediction accuracy or transmitting data packets. The above behaviors explain the performance characteristic of OSSLMS and HLMS that their duty cycle results are inversely proportional with the value of the wakeup interval. The results of the proposed model are less impacted by the value of the wakeup interval since the proposed model requires only one sensor for each ICC to be active and operate at the duty-cycled mode.

TABLE 3. **Numerical result with highly interference scenario.**

	Proposed system	OSSLMS	HLMS
Latency	$0.54s$	$2.67s$	$1.98s$
Request successful rate	99.2%	88.6%	91.4%

G. SERVICE AVAILABILITY IN HIGH INTERFERENCE SCENARIO

We create multiple application requests for sensing services under a high interference scenario as described in [56]. Obtained average latency and request successful rate results are presented in table 3 which indicates the advantages of the proposed system in term of increasing the service availability. As the data are available in the sensor cloud and most of application requests can be satisfied by predicted values of IPVs in the sensor cloud, the proposed system helps improve the response latency and request successful rate significantly compared to OSSLMS and HLMS even in high interference scenario.

V. CONCLUSIONS

This paper presents an energy efficient integration model for WSNs and the sensor cloud, in which IPVs and IPDs are decoupled to enable IPVs to provide sensing services regardless of how information objects are produced. The purpose is to enable IPVs to be able to predict sensing data of IPDs accurately and use those data for providing sensing services even when IPDs sleep. We design an efficient interactive prediction scheme for IPVs and IPDs so that IPVs predict sensing data under control by active IPDs based on the requirement of applications. Obtained results show that the proposed model achieves significant improvements in terms of data transmission suppression ratio, energy efficiency, and response latency, compared to the existing schemes. In addition, the model is designed for the sensor cloud to support multiple applications simultaneously. Most of IPDs in the proposed model are allowed to sleep deeply, instead of just suppressing data transmissions like the existing schemes.

ACKNOWLEDGMENT

This paper was presented at the 2017 IEEE International Conference on Communications [1].

REFERENCES

[1] T. Dinh and Y. Kim, “Information centric sensor-cloud integration: An efficient model to improve wireless sensor networks’ lifetime,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.
[2] M. Wu, L. Tan, and N. Xiong, “Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications,” *Inf. Sci.*, vol. 329, pp. 800–818, Feb. 2016.
[3] J. Pardo, F. Zamora-Martinez, and P. Botella-Rocamora, “Online learning algorithm for time series forecasting suitable for low cost wireless sensor networks nodes,” *Sensors*, vol. 15, no. 4, pp. 9277–9304, 2015.
[4] L. Tan and M. Wu, “Data reduction in wireless sensor networks: A hierarchical LMS prediction approach,” *IEEE Sensors J.*, vol. 16, no. 6, pp. 1708–1715, Mar. 2016.
[5] C. Sarkar, V. S. Rao, and R. V. Prasad, “No-sense: Sense with dormant sensors,” in *Proc. 12th Nat. Conf. Commun. (NCC)*, Feb. 2014, pp. 1–6.
[6] S. He, J. Chen, D. K. Y. Yau, and Y. Sun, “Cross-layer optimization of correlated data gathering in wireless sensor networks,” *IEEE Trans. Mobile Comput.*, vol. 11, no. 11, pp. 1678–1691, Nov. 2012.
[7] E. Karasabun, I. Korpeoglu, and C. Aykanat, “Active node determination for correlated data gathering in wireless sensor networks,” *Comput. Netw.*, vol. 57, no. 5, pp. 1124–1138, Apr. 2013.
[8] T. Dinh and Y. Kim, “An efficient interactive model for on-demand sensing-as-a-servicesof sensor-cloud,” *Sensors*, vol. 16, no. 7, p. 992, 2016.
[9] T. Dinh and Y. Kim, “A novel location-centric IOT-cloud based on-street car parking violation management system in smart cities,” *Sensors*, vol. 16, no. 6, p. 810, 2016.
[10] M. Fazio and A. Puliafito, “Cloud4sens: A cloud-based architecture for sensor controlling and monitoring,” *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 41–47, Mar. 2015.
[11] S. Madria, V. Kumar, and R. Dalvi, “Sensor cloud: A cloud of virtual sensors,” *IEEE Softw.*, vol. 31, no. 2, pp. 70–77, Mar. 2014.
[12] G. M. Dias, B. Bellalta, and S. Oechsner, “A survey about prediction-based data reduction in wireless sensor networks,” *ACM Comput. Surv.*, vol. 49, no. 3, pp. 58:1–58:35, Nov. 2016.
[13] N. Dinh and Y. Kim, “Potential of information-centric wireless sensor and actor networking,” in *Proc. Int. Conf. Comput., Manage. Telecommun. (ComManTel)*, Jan. 2013, pp. 163–168.
[14] N.-T. Dinh and Y. Kim, “Information-centric dissemination protocol for safety information in vehicular ad-hoc networks,” *Wireless Netw.*, vol. 23, no. 5, pp. 1359–1371, 2017.
[15] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named content,” in *Proc. 5th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, 2009, pp. 1–12.
[16] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wahlisch, “Information centric networking in the IoT: Experiments with NDN in the wild,” in *Proc. 1st Int. Conf. Inf.-Centric Netw. (ICN)*, 2014, pp. 77–86.

[17] M. Amadeo *et al.*, "Information-centric networking for the Internet of Things: Challenges and opportunities," *IEEE Netw.*, vol. 30, no. 2, pp. 92–100, Mar. 2016.
[18] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.
[19] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010.
[20] S. Madden. *Intel Berkeley Research Lab Data*. Accessed: Dec. 24, 2018. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html
[21] T. Dinh, Y. Kim, and H. Lee, "A location-based interactive model of Internet of Things and cloud (IoT-cloud) for mobile cloud computing applications," *Sensors*, vol. 17, no. 3, p. 489, 2017.
[22] T. Dinh, Y. Kim, T. Gu, and A. V. Vasilakos, "L-MAC: A wake-up time self-learning MAC protocol for wireless sensor networks," *Comput. Netw.*, vol. 105, pp. 33–46, Aug. 2016.
[23] T. Dinh and Y. Kim, "An efficient sensor-cloud interactive model for on-demand latency requirement guarantee," in *Proc. IEEE ICC*, May 2017, pp. 1–6.
[24] N. Dinh and Y. Kim, "An efficient on-demand latency guaranteed interactive model for sensor-cloud," *IEEE Access*, vol. 6, pp. 36906–36918, 2018, doi: 10.1109/ACCESS.2018.2879811.
[25] Y. Park, H. Yang, T. Dinh, and Y. Kim, "Design and implementation of a container-based virtual client architecture for interactive digital signage systems," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 7, pp. 1–15, 2017.
[26] *Network Functions Virtualisation (NFV); Architectural Framework*, document ETSI GS NFV 002 V1.1.1, ETSI, 2013.
[27] A.-V. Vu, T. Dinh, and Y. Kim, "Modeling of service function chaining in network function virtualization environment," in *Proc. KICS Summer Conf.*, Jeju, South Korea, 2016, pp. 1–2.
[28] G. Fortino, D. Parisi, V. Pirrone, and G. Di Fatta, "BodyCloud: A SaaS approach for community body sensor networks," *Future Gener. Comput. Syst.*, vol. 35, pp. 62–79, Jun. 2014.
[29] G. Fortino, G. Di Fatta, M. Pathan, and A. V. Vasilakos, "Cloud-assisted body area networks: State-of-the-art and future challenges," *Wireless Netw.*, vol. 20, no. 7, pp. 1925–1938, 2014.
[30] S. Ghanavati, J. Abawajy, and D. Izadi, "An alternative sensor cloud architecture for vital signs monitoring," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 2827–2833.
[31] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, and M. Aktas, "Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jul. 2015, pp. 285–292.
[32] S. Misra, A. Singh, S. Chatterjee, and M. S. Obaidat, "Mils-cloud: A sensor-cloud-based architecture for the integration of military tri-services operations and decision making," *IEEE Syst. J.*, vol. 10, no. 2, pp. 628–636, Jun. 2016.
[33] L. Neto, J. Reis, R. Silva, and G. Gonçalves, "Sensor SelComp, a smart component for the industrial sensor cloud of the future," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2017, pp. 1256–1261.
[34] Y. Lyu, F. Yan, Y. Chen, D. Wang, Y. Shi, and N. Agoulmine, "High-performance scheduling model for multisensor gateway of cloud sensor system-based smart-living," *Inf. Fusion*, vol. 21, pp. 42–56, Jan. 2015.
[35] S. Goel and T. Imielinski, "Prediction-based monitoring in sensor networks: Taking lessons from MPEG," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 5, pp. 82–98, Oct. 2001.
[36] C. Liu, K. Wu, and M. Tsao, "Energy efficient information collection with the ARIMA model in wireless sensor networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 5, Dec. 2005, p. 2474.
[37] A. Jain, E. Y. Chang, and Y.-F. Wang, "Adaptive stream resource management using Kalman filters," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2004, pp. 11–22.
[38] J.-K. Min and C.-W. Chung, "EDGES: Efficient data gathering in sensor networks using temporal and spatial correlations," *J. Syst. Softw.*, vol. 83, no. 2, pp. 271–282, 2010.
[39] J. Kho, A. Rogers, and N. R. Jennings, "Decentralized control of adaptive sampling in wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 5, no. 3, pp. 19:1–19:35, Jun. 2009.
[40] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco, "What does model-driven data acquisition really achieve in wireless sensor networks?" in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2012, pp. 85–94.
[41] Y.-A. Le Borgne, S. Santini, and G. Bontempi, "Adaptive model selection for time series prediction in wireless sensor networks," *Signal Process.*, vol. 87, no. 12, pp. 3010–3020, 2007.
[42] *Machine-to-Machine Communications (M2M); Functional Architecture*, document ETSI-TS 102 690 V2.1.1, ETSI, 2013.
[43] C. Zhu, X. Li, V. C. M. Leung, L. T. Yang, E. C.-H. Ngai, and L. Shu, "Towards pricing for sensor-cloud," *IEEE Trans. Cloud Comput.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/7809146, doi: 10.1109/TCC.2017.2649525.
[44] S. Chatterjee, R. Ladia, and S. Misra, "Dynamic optimal pricing for heterogeneous service-oriented architecture of sensor-cloud infrastructure," *IEEE Trans. Services Comput.*, vol. 10, no. 2, pp. 203–216, Mar. 2017.
[45] *OpenStack*. Accessed: Dec. 24, 2018. [Online]. Available:https://www.openstack.org/
[46] C. Liang, F. R. Yu, and X. Zhang, "Information-centric network function virtualization over 5G mobile wireless networks," *IEEE Netw.*, vol. 29, no. 3, pp. 68–74, May 2015.
[47] J. Ren *et al.*, "On the deployment of information-centric network: Programmability and virtualization," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2015, pp. 690–694.
[48] N.-T. Dinh and Y. Kim, "An efficient on-demand latency guaranteed interactive model for sensor-cloud," *IEEE Access*, vol. 6, pp. 36906–36915, 2018.
[49] C.-P. Hoang, N.-T. Dinh, and Y. Kim, "An extended virtual network functions manager architecture to support container," in *Proc. Int. Conf. Inf. Sci. Syst. (ICISS)*, 2018, pp. 173–176.
[50] S. He, J. Chen, X. Li, X. Shen, and Y. Sun, "Leveraging prediction to improve the coverage of wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 4, pp. 701–712, Apr. 2012.
[51] S. S. Haykin, *Adaptive Filter Theory*, 4th ed. London, U.K.: Pearson Education India, 2005.
[52] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*. Hoboken, NJ, USA: Wiley, 2012.
[53] S. Santini and K. Römer, "An adaptive strategy for quality-based data reduction in wireless sensor networks," in *Proc. 3rd Int. Conf. Netw. Sens. Syst. (INSS)*, Jun. 2006, pp. 29–36.
[54] Y. Park, N.-T. Dinh, and Y. Kim, "A network monitoring system in 6LoWPAN networks," in *Proc. 4th Int. Conf. Commun. Electron. (ICCE)*, Aug. 2012, pp. 69–73.
[55] T. Dinh and T. Gu, "A novel metric for opportunistic routing in heterogenous duty-cycled wireless sensor networks," in *Proc. IEEE 23rd Int. Conf. Netw. Protocols (ICNP)*, Nov. 2015, pp. 224–234.
[56] T. Dinh, Y. Kim, T. Gu, and A. V. Vasilakos, "An adaptive low-power listening protocol for wireless sensor networks in noisy environments," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2162–2173, Sep. 2018.

NGOC-THANH DINH received the M.Sc. and Ph.D. degrees in electronic and telecommunication from Soongsil University, and the Ph.D. degree from the School of Computer Science and IT, Royal Melbourne Institute of Technology University. He is currently an Assistant Professor with the Department of IT Convergence, Soongsil University. His current research interests include the Internet of Things and cloud computing, 5G networking, and next-generation networks. His publications appear in top journals, such as the IEEE INTERNET OF THINGS, the IEEE TRANSACTIONS ON BIG DATA, and the IEEE SYSTEMS JOURNAL, and top conferences, such as the IEEE ICNP and ICC. He serves as a TPC member for many leading conferences and a member for journals, such as the IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE SENSORS, and *Computer Networks*.

YOUNGHAN KIM received the B.S. degree from Seoul National University and the M.Sc. and Ph.D. degrees in electrical engineering from KAIST. He is currently a Full Professor with the Department of IT Convergence, Soongsil University. He is also an Executive Director of the Korea Information and Communications Society and the President of the Open Standards and Internet Association.