Multiobjective Communication Optimization for Cloud-integrated Body Sensor Networks

Dũng H. Phan*, Junichi Suzuki*, Shingo Omura†, Katsuya Oba† and Athanasios V. Vasilakos‡
*Department of Computer Science
University of Massachusetts, Boston
Boston, MA 02125-3393, USA
Email: {phdung,jxs}@cs.umb.edu
†OGIS International, Inc.
San Mateo, CA 94402, USA
Email: {omura,oba}@ogis-international.com
‡Computer Science Department
Kuwait University
P.O. Box 5969, Safat 13060, Kuwait
Email: th.vasilakos@gmail.com

***Abstract*—This paper focuses on push-pull hybrid communication in a cloud-integrated sensor networking architecture, called Sensor-Cloud Integration Platform as a Service (SC-iPaaS). SC-iPaaS consists of three layers: sensor, edge and cloud layers. The sensor layer consists of wireless body sensor networks, each of which operates several sensors for a homebound patient for a remote physiological and activity monitoring. The edge layer consists of sink nodes that collect sensor data from sensor nodes in the sensor layer. The cloud layer hosts cloud applications that obtain sensor data through sink nodes in the edge layer. This paper formulates an optimization problem for SC-iPaaS to seek the optimal data transmission rates for individual sensor and edge nodes and solves the problem with respect to multiple objectives (e.g., data yield, bandwidth consumption and energy consumption) subject to given constraints. This paper sets up a simulation environment that performs remote multi-patient monitoring with five on-body sensors including ECG, pulse oximeter and accelerometer per a patient. Simulation results demonstrate that the proposed optimizer successfully seeks Pareto-optimal data transmission rates for sensor/sink nodes against data request patterns placed by cloud applications. The results also confirm that the proposed optimizer outperforms an existing well-known optimization algorithm.**

I. Introduction

This paper proposes a cloud-integrated architecture for wireless sensor networks and evaluates a communication optimizer for the architecture. The proposed architecture, called Sensor-Cloud Integration Platform as a Service (SC-iPaaS) is a three-tier communication architecture that seamlessly integrates the *sensor*, *edge* and *cloud* layers. The sensor layer consists of sensor nodes embedded in the physical environment. The edge layer consists of sink nodes that collect sensor data from sensor nodes in the physical environment. The cloud layer consists of cloud computing environments that host *virtual sensors*, which are virtualized counterparts (or software counterparts) of physical sensors in the sensor layer. Virtual sensors collect sensor data from sink nodes in the edge layer and store those data for future use. Clouds also host cloud applications that obtain sensor data from virtual sensors and aid users to monitor physical phenomena, events and processes in the physical environment.

SC-iPaaS performs *push-pull hybrid communication* between its layers. Individual sensor nodes periodically transmit (or push) sensor data to sink nodes, which in turn forward (or push) incoming sensor data periodically to virtual sensors in clouds. When a virtual sensor does not have sensor data that a cloud application requires, it obtains (or pulls) that data from a sink node or a sensor node. This push-pull communication scheme is intended to make as much sensor data as possible readily available for cloud applications by taking advantage of push communication while allowing virtual sensors to pull any missing data anytime in an on-demand manner.

An example application of SC-iPaaS is remote physiological and activity monitoring in pervasive healthcare for homebound patients [1]–[3]. This application assumes per-patient wireless networks of on-body and/or in-body sensors for, for example, heart rate, blood pressure, oxygen saturation, body temperature, respiratory rate, blood coagulation, galvanic skin response and fall detection. Those sensors are wirelessly connected via single-hop or multi-hop paths to a dedicated per-patient device or a patient's computer (e.g., cell phone, tablet machine or laptop computer) that serves as a sink node. Real-time physiological and activity sensor data are periodically pushed to virtual sensors in clouds so that clinicians, hospital nurses and visiting nurses can share the data for clinical observation and intervention. When an anomaly is found in physiological/activity sensor data, clinical staff may pull extra data in a higher temporal resolution to better understand a patient's medical condition. Given a sufficient amount of data, they may order/perform clinical interventions, dispatch ambulances or notify family members of patients.

This paper focuses on a communication optimization problem in SC-iPaaS and solves it with an evolutionary optimization algorithm. The problem is to seek the optimal data transmission rate for each sensor node and sink node with respect to multiple optimization objectives such as sensor data yield (sensor data availability) for cloud applications, band-

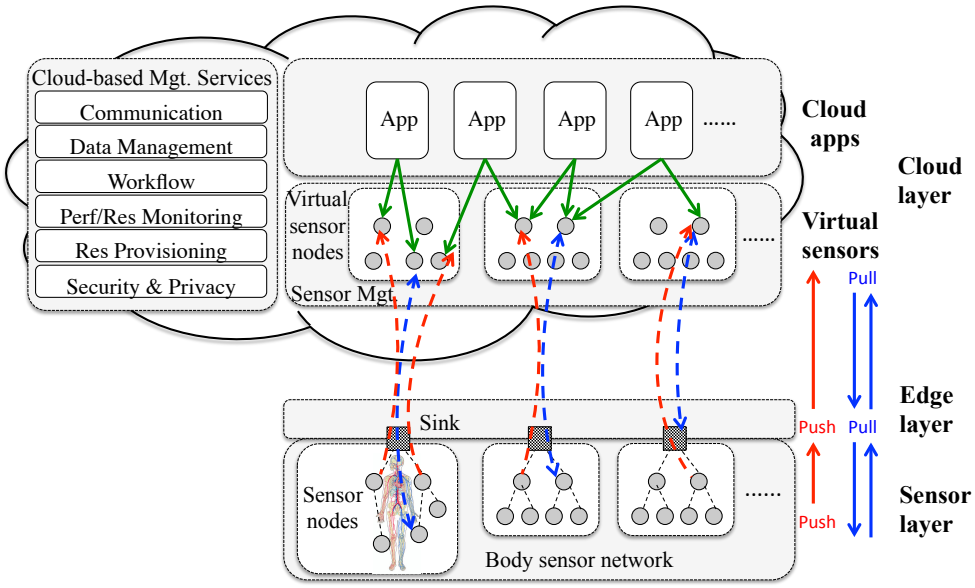


Fig. 1. A Push-Pull Hybrid Communication Architecture for Cloud-integrated Sensor Networks

width consumption between the cloud layer and the edge layer and energy consumption of sensor nodes in the sensor layer. This paper sets up a simulation environment that performs remote multi-patient monitoring with five on-body sensors including ECG, pulse oximeter and accelerometer per a patient. Simulation results demonstrate that the proposed optimization algorithm successfully seeks Pareto-optimal communication configurations (i.e., data transmission rates for sensor/sink nodes) against data request patterns placed by cloud applications. The results also confirm that the proposed algorithm outperforms an existing well-known optimization algorithm.

II. An Overview of SC-iPaaS

Figure 1 shows an architectural overview of SC-iPaaS. SC-iPaaS consists of the following three tiers.

Sensor Layer: operates one or more wireless networks of stationary sensor nodes embedded in the physical environment. Each network is assumed to be heterogeneous; it has different types of sensor devices such as air temperature sensors, humidity sensors and barometric pressure sensors. Sensor nodes are battery-operated or solar-powered; they have limited energy supplies. In each sensor network, nodes form a particular topology (e.g., tree, star or mesh topology). In Figure 1, sensor networks use a tree topology. Nodes periodically read sensors and transmit (or push) sensor data to a special node, called sink node, on a hop-by-hop manner through a given network topology. Different sensor nodes have different data transmission rates.

Edge Layer: is a collection of sink nodes, each of which participates in a certain sensor network and receives sensor data periodically from individual nodes in the network. Each sink node stores incoming sensor data in its memory space and then transmits (or pushes) them periodically to the cloud layer. It maintains the mappings between physical sensors and virtual sensors. In other words, it knows the origins and destinations of sensor data. Different sink nodes have different data transmission rates. A sink node's data transmission rate can be different from the ones of sensor nodes in the same network. Note that each sensor network operates one or more sink nodes. In Figure 1, one sink node is operated in each sensor network. Depending on application domains, sink nodes may have limited energy supplies through batteries and solar panels or they may have infinite energy supplies.

In addition to pushing sensor data to a virtual sensor, each sink node receives a "pull" request from a virtual sensor when the virtual sensor does not have data that a cloud application(s) requires (Figure 1). If the sink node has the requested data in its memory, it returns that data. Otherwise, it issues a pull request to a sensor node that is responsible for the requested data. Upon receiving the pull request, the sensor node reads a sensor and returns sensor data.

Cloud Layer: operates on one or more clouds to host end-user applications and management services for the applications. Applications are operated on virtual machines in clouds. Users are assumed to place continuous sensor data queries on virtual sensors via cloud applications in order to monitor the physical environment. If a virtual sensor already has data that an application queries, it returns that data. If a query does not match, the virtual sensor issues a pull request and sends it to a sink node. Each query is assumed to have a relative time window within which an application requires particular sensor data. While push communication carries out one-way upstream travel of sensor data, pull communication incurs a round trip for requesting sensor data and receiving that data (Figure 1).

Cloud-based management services offer common functionalities to implement and operate applications (Figure 1). This paper focuses on the following two services.

- *Sensor manager*: virtualizes physical heterogeneous sensors in a unified way by abstracting away their low-level operational details. Cloud applications always access physical sensors through virtual sensors; for example, collecting sensor data with a pull request and

sending control signals (e.g., turning on/off sensors and setting data transmission rates).

- *Communication manager*: is responsible for push-pull hybrid communication between different layers. It is assumed to operate on top of certain publish/subscribe communication middleware such as TinyDDS [4]. A key component in this manger is *communication optimizer*, which this paper focuses on to seek the optimal data transmission rates for sensor and sink nodes with respect to multiple optimization objectives.

III. Communication Optimization Problem in SC-iPaaS

This section describes an optimization problem to seek the optimal data transmission rates for sensor and sink nodes in SC-iPaaS. The following notations are used to state the optimization problem.

- $S = \{s_1, s_2, ..., s_i, ..., s_M\}$ is the set of M sensor nodes in sensor networks. ν_{s_i} denotes the data transmission rate for the i-th sensor node (s_i) to push sensor data to its corresponding sink node. The rate is measured as the number of sensor data transmitted per a unit time. d_i indicates the size of single sensor data that s_i generates and transmits to a sink node. h_i denotes the shortest logical distance (i.e., hop count) from s_i to its corresponding sink node.
- ν_{e_i} denotes the data transmission rate for a sink node to push sensor data receiving from the i-th sensor node (s_i). ν_{e_i} and ν_{s_i} are not necessarily equal ($\nu_{e_i} \leq \nu_{s_i}$).
- W indicates a relative time window that SC-iPaaS considers to monitor sensor data requests from cloud applications and compute its communication performance with respect to optimization objectives.
- $R_i = \{r_{i1}, r_{i2}, ..., r_{ij}, ..., r_{i|R_i|}\}$ is the set of all sensor data requests that cloud applications issue to the virtual counterpart of s_i (s'_i) during the time period of W in the past. r_{ij} denotes the j-th sensor data request from a cloud application to the i-th virtual sensor s'_i during the time period of W in the past. Each request is characterized by its time stamp (t_{ij}) and time window (w_{ij}). It requests all sensor data available in the time interval $[t_{ij} - w_{ij}, t_{ij}]$. If s'_i has at least one data in $[t_{ij} - w_{ij}, t_{ij}]$, it returns those data to a cloud application. Otherwise, it issues a pull request to a sink node.
- $R^e_i \in R_i$ is the set of sensor data requests for which the virtual sensor s'_i has no data. This means that $|R^e_i|$ indicates the number of pull requests that s'_i issues to a sink node. In other words, $R_i \setminus R^e_i$ indicates the requests that s'_i can fulfill.
- $R^s_i \in R^e_i \in R_i$ is the set of sensor data requests for which the sink node for s_i do not have data. This means that $|R^s_i|$ indicates the number of pull requests that the sink node issues to s_i. In other words, $R^e_i \setminus R^s_i$ indicates the requests that the sink node can fulfill.

This paper considers three optimization objectives: bandwidth consumption between the edge and cloud layers (f_B), energy consumption of physical sensors (f_E) and data yield for cloud applications (f_D). The first two objectives are to be minimized while the third is to be maximized.

The bandwidth consumption objective (f_B) is defined as the total amount of data transmitted per a unit time between the edge and cloud layers. This objective impacts the payment for bandwidth consumption based on a cloud operator's pay-per-use billing scheme. It also impacts the lifetime of sink nodes if they are battery-operated or solar-powered. f_B is computed as follows.

$$f_B = \sum_{i=1}^{M} (\nu_{ei} \times d_i) + \frac{1}{W} \sum_{i=1}^{M} \sum_{r_{ij} \in R^e_i} (\phi_{ij} \times d_i + d_r) \quad (1)$$

The first and second terms of the equation indicate the bandwidth consumption by one-way push communication from the edge layer to the cloud layer and two-way pull communication between the cloud and edge layers, respectively. ϕ_{ij} denotes the number of sensor data that the request r_{ij} can collect in the time interval $[t_{ij} - w_{ij},\ t_{ij}]$. d_r indicates the size of a single pull request from the cloud layer to the edge layer. It is constant for all sensor nodes.

The energy consumption objective (f_E) is defined as the total amount of energy that sensor nodes consume for data transmissions during the time period of W. This objective impacts the lifetime of sensor nodes and sensor networks. It is computed as follows.

$$f_E = \sum_{i=1}^{M} (h_i \times e_t \times d_i \times \nu_{si} \times W) + \sum_{i=1}^{M} \sum_{r_{ij} \in P_{si}} \left(h_i \times e_t \times (d_i + d'_r) \right) \quad (2)$$

The first and second terms indicate the energy consumption by one-way push communication from the sensor layer to the edge layer and two-way pull communication between the edge layer and the sensor layer, respectively. e_t denotes the amount of energy that a unit amount of data consumes to travel from a sensor node to its neighboring node. d'_r denotes the size of a single pull request from the edge layer to the sensor layer. e_t and d_r are constant for all sensor nodes.

The data yield objective (f_Y) is defined as the total amount of data that cloud applications gather for their users. This objective impacts the informedness and situation awareness for application users. It is computed as follows.

$$f_Y = \sum_{i=1}^{M} \sum_{r_{ij} \in R_i} \phi_{ij} \quad (3)$$

In SC-iPaaS, optimization objectives conflict with each other. For example, the data yield objective conflicts with the other two objectives. Maximizing data yield means increasing data transmission rates for sensor and sink nodes. This increases bandwidth consumption and energy consumption. Similarly, the energy consumption objective conflicts with the data yield objective. Minimizing energy consumption means reducing data transmission rates for sensor nodes. This can

reduce data yield. Given these conflicting objectives, this paper seeks the optimal trade-off (i.e., Pareto-optimal) configurations for data transmission rates in SC-iPaaS.

SC-iPaaS considers two constraints in its optimization process. The first constraint (C_E) is the upper limit for energy consumption (f_E):

$$f_E < C_E \tag{4}$$

The constraint violation in energy consumption (g_E) is computed as follows where $I = 1$ if $f_E > C_E$; otherwise $I = 0$.

$$g_E = I \times (f_E - C_E) \tag{5}$$

The second constraint (C_Y) is the lower limit for data yield (f_Y):

$$f_Y > C_Y \tag{6}$$

The constraint violation in data yield (g_Y) is computed as follows where $I = 1$ if $f_E < C_E$; otherwise $I = 0$.

$$g_Y = I \times (C_Y - f_Y) \tag{7}$$

IV. Communication Optimizer in SC-iPaaS

SC-iPaaS leverages a novel evolutionary multiobjective optimization algorithm (EMOA) for its communication optimizer. The proposed EMOA is intended to search Pareto-optimal solutions that are equally distributed in the objective space because there exits no single optimal solution under conflicting objectives but rather a set of alternative solutions of equivalent quality. Therefore, it can produce both *extreme* data transmission configurations (e.g., the one exhibiting high data yield and high energy consumption) and *balanced* configurations (e.g., the one exhibiting intermediate data yield and energy consumption) at the same time. Given a set of heuristically-approximated Pareto-optimal solutions, an SC-iPaaS operator can examine the trade-offs among them and make a well-informed decision to choose one of them, as the data transmission configuration, according to his/her preferences and priorities. For example, an SC-iPaaS operator can examine how he/she can data yield for energy consumption and determine a particular data transmission configuration that achieves a desirable/comfortable balance of data yield and energy consumption.

A. Individual Representation

The proposed EMOA iteratively evolves the population of solution candidates, called *individuals*, with several operators (e.g., crossover, mutation and selection operators) toward the Pareto-optimal solutions in the objective space. In SC-iPaaS, each individual (solution candidate) represents a particular data transmission configuration, which is a set of data transmission rates for all sensor and sink nodes (Figure 2).

ν_{e1}	ν_{e2}	ν_{e3}	....	ν_{eM-1}	ν_{eM}
ν_{s1}	ν_{s2}	ν_{s3}	....	ν_{sM-1}	ν_{sM}

Fig. 2. Individual Representation

B. R2 Indicator

The proposed EMOA utilizes a quality indicator, called the R2 indicator [5], in order to assess the quality (or optimality) of an individual. The R2 indicator was originally proposed to assess the relative quality of two sets of individuals [5]. Assuming the standard weighted Tchebycheff function with a particular reference point z^*, the indicator can be used to assess the quality of a individual set ($\mathcal{A}$) against z^* [6], [7]:

$$R_2(\mathcal{A}, \mathcal{V}, z^*) = \sum_{v \in \mathcal{V}} \Big(p(v) \times \min_{a \in \mathcal{A}} \{ \max_{1 \le j \le m} v_j | z_j^* - a_j | \} \Big) \tag{8}$$

$\mathcal{V}$ denotes a set of *weight vectors*. Each weight vector $v = (v_1, ..., v_{|\mathcal{V}|}) \in \mathcal{V}$ is placed in the m-dimensional objective space. p denotes a probability distribution on $\mathcal{V}$. Weight vectors are often chosen uniformly distributed in the objective space [6]–[9]. In this case, the R2 indicator is described as:

$$R_2(\mathcal{A}, \mathcal{V}, z^*) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \min_{a \in \mathcal{A}} \{ \max_{1 \le j \le m} v_j | z_j^* - a_j | \} \tag{9}$$

A utopian point is usually used as the reference point z^* [6]–[9]. A utopian point is a point that is never dominated by any feasible solutions in the objective space. For example, it is $(0, 0)$ in a two-dimensional objective space where each objective value is greater than or equal to 0.

A lower R2 value indicates that an individual set $\mathcal{A}$ is closer to the reference point. $R2(\{x\}, \mathcal{V}, z^*) = 0$ when an individual $x \in \mathcal{S}$ is positioned on the reference point. The R2 indicator possesses a desirable property of *weak monotonicity*. When an individual $x \in \mathcal{S}$ dominates another individual $y \in \mathcal{S}$ in the objective space, $R2(\{x\}, \mathcal{V}, z^*) \le R2(\{y\}, \mathcal{V}, z^*)$.

C. Optimization Process

Algorithm 1 shows the evolutionary optimization process in SC-iPaaS. First, uniformly distributed weight vectors $\mathcal{V}$ are generated (Equation 9). This vector generation is executed offline only once.

In the first generation ($g = 0$), μ individuals are randomly generated as the initial population $\mathcal{P}_0$ (Line 2). In each generation (g), a pair of individuals, called parents (p_1 and p_2), are chosen from the current population $\mathcal{P}_g$ with a binary tournament operator (Lines 7 and 8). This operator randomly draws two individuals from $\mathcal{P}_g$, compares them with the *constrained binary R2 indicator*, and selects a superior one as a parent. The constrained binary R2 indicator accepts two individuals (x and y) and determines which one is superior:

- if both x and y are *feasible*, which means both never violate all of optimization constraints,

Algorithm 1 Optimization Process in SC-iPaaS

1: $g = 0$
2: Generate uniformly distributed weight vectors $\mathcal{V}$
3: $\mathcal{P}_g$ = initializePopulation(μ)
4: **while** g $< g_{max}$ **do**
5: $\mathcal{O}_g = \emptyset$
6: **while** $|\mathcal{O}_g| < \mu$ do **do**
7: p_1 =binaryTounament($\mathcal{P}_g$)
8: p_2 =binaryTounament($\mathcal{P}_g$)
9: **if** random() $\leq P_c$ **then**
10: $\{o_1, o_2\}$ = crossover(p_1, p_2)
11: **if** random() $\leq P_m$ **then**
12: o_1 = mutation(o_1)
13: **end if**
14: **if** random() $\leq P_m$ **then**
15: o_2 = mutation(o_2)
16: **end if**
17: $\mathcal{O}_g = \{o_1, o_2\} \cup \mathcal{O}_g$
18: **end if**
19: **end while**
20: $\mathcal{R}_g = \mathcal{P}_g \cup \mathcal{O}_g$
21: Calculate the fitness of each individual $x_i \in \mathcal{R}_g$ as: $F(x_i) = \sum_{y_i \in \mathcal{R}_g \setminus \{x_i\}} -e^{-I^c_{R2}(y_i, x_i)/\kappa}$
22: **while** $|\mathcal{R}_g| > \mu$ **do**
23: $x^* = \arg\min_{x_i \in \mathcal{R}_g} F(x_i)$
24: $\mathcal{R}_g = \mathcal{R}_g \setminus \{x^*\}$
25: Update the fitness of each individual $x_i \in \mathcal{R}_g$ as: $F(x_i) = F(x_i) + e^{-I^c_{R2}(x^*, x_i)/\kappa}$
26: **end while**
27: $g = g + 1$
28: **end while**

$$I^c_{R_2}(x, y) = R_2(\{x\}, \mathcal{V}, z^*) - R_2(\{x \cup y\}, \mathcal{V}, z^*) \tag{10}$$

R_2 values are computed in a three-dimensional objective space because this paper considers three optimization objectives ($m = 3$ in Equation 9).

- if x is feasible and y is not,

$$I^c_{R_2}(x, y) = 0 \tag{11}$$

$$I^c_{R_2}(y, x) = R_2(\{x\}, \mathcal{V}, z^*) \tag{12}$$

- if both x and y are infeasible, which means both violate at least one optimization constraints,

$$I^c_{R_2}(x, y) = R^c_2(\{x\}, \mathcal{V}, z^*) - R^c_2(\{x \cup y\}, \mathcal{V}, z^*) \tag{13}$$

R^c_2 is a R_2 value that is computed in the constraint space based on Equation 9. The constraint space is a space whose axes (or dimension) represents optimization constraints. This paper uses a two-dimensional constraint space because two constraints are considered (Section III).

The constrained binary R2 indicator determines that x is superior against y if $I^c_{R_2}(x, y) < I^c_{R_2}(y, x)$. If $I^c_{R_2}(x, y) = I^c_{R_2}(y, x)$, the binary tournament operator chooses either one at random as a parent.

With the crossover rate P_c, two parents reproduce two offspring with the SBX (self-adaptive simulated binary crossover) operator [10] (Line 10). Polynomial mutation [11] is performed on each offspring with the mutation rate P_m (Lines 11 to 16). Parent selection, crossover and mutation operators are repeatedly executed on $\mathcal{P}_g$ until μ offspring are reproduced (i.e., until $|O_g| = \mu$). The offspring ($\mathcal{O}_g$) are combined with the population $\mathcal{P}_g$ to form $\mathcal{R}_g$ ($|R_g| = 2\mu$), which is a pool of candidates for the next-generation individuals (Line 20).

Environmental selection follows offspring reproduction (Lines 21 to 26). In Line 21, the fitness of each individual in $\mathcal{R}_g$ is calculated by applying the individual's I^c_{R2} value to an exponential amplification function. Then, the worst individual (i.e., the one with the lowest fitness) is removed from $\mathcal{R}_g$ (Lines 23 and 24). In Line 25, fitness is recalculated for each of the remaining individuals in $\mathcal{R}_g$. This individual removal process is repeated until $|\mathcal{R}_g| = \mu$. The μ individuals form the population used in the next generation ($\mathcal{P}_{g+1}$).

V. Simulation Evaluation

This section evaluates the proposed optimizer in SC-iPaaS through simulations.

A. Simulation Configurations

A simulation environment is set up to perform remote physiological and activity monitoring for 10 patients. Each patient carries a sink node and wears five different sensors. One sink node and five sensor nodes form a multi-hop wireless body sensor network (Figure 1). Cloud applications are simulated to issue 10,000 sensor data requests during a day. Those requests are uniformly distributed over 50 virtual sensors.

The proposed optimizer is configured with a set of parameters shown in Table I.

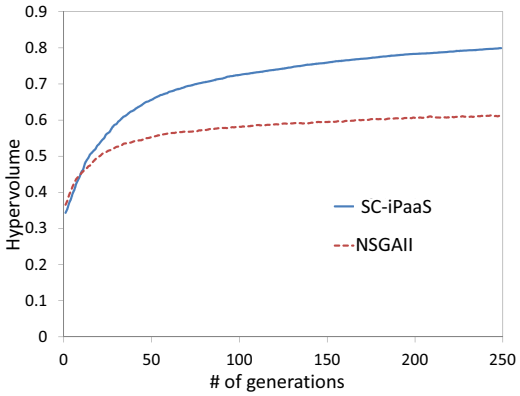
TABLE I. Simulation Configurations

Parameter	Value
Total # of sensors (M)	50
Total # of data requests	10,000
Simulation time (W)	1 day
# of weight vectors ($\|\mathcal{V}\|$)	100
Reference point (z^*)	(0, 0, 0)
Population size (μ)	100
Max # of generations (g_{max})	250
Crossover rate (P_c)	0.9
Mutation rate (P_m)	0.01
Amplification coefficient (κ)	0.005

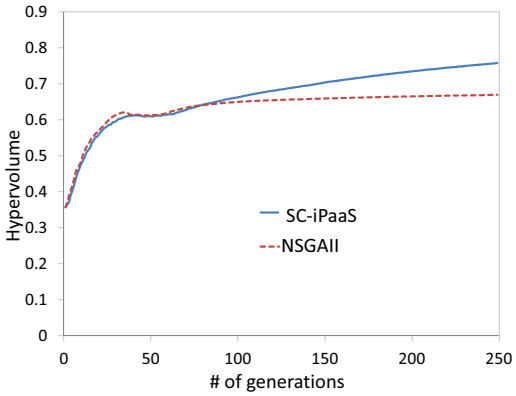
TABLE II. Configurations for Sensors and Sensor Data Requests

Sensor type	Quantity	Data size (d_i) (Bytes)	Request time window (w_{ij}) (Seconds)
ECG	10	128	$\mathcal{N}(60, 10^2)$
Pulse Oximeter	10	15	$\mathcal{N}(200, 50^2)$
Accelerometer	10	100	$\mathcal{N}(40, 10^2)$
Body temperature	10	10	$\mathcal{N}(600, 100^2)$
Blood pressure	10	10	$\mathcal{N}(300, 80^2)$

Table II shows five types of sensors used in simulations. 10 sensors are used for each type. The size of each ECG data



(a) Hypervolume Measure over Generations (without constraints)



(b) Hypervolume Measure over Generations (with constraints)

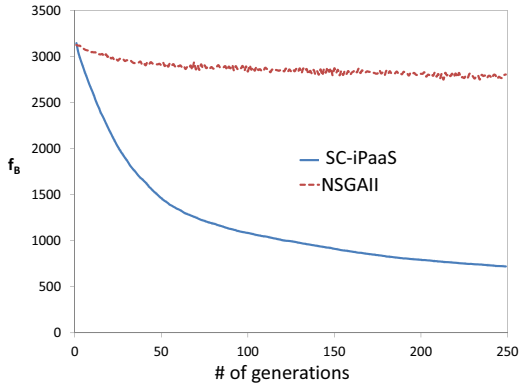
Fig. 3. Average Objective Values over Generations

is 128 bytes. The time window for each ECG data request is randomly generated following a normal distribution with the mean of 60 seconds and the standard deviation of 10 seconds.

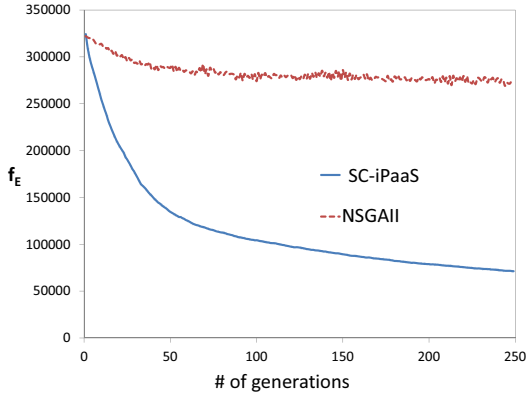
B. Simulation Results

Figure 3(a) shows how individuals evolve as the number of generations grows when no constraints are specified: $C_Y = 0$ and $C_E = \infty$. It uses the hypervolume metric [12], which measures the union of volumes that individuals dominates in the objective space. Thus, the hypervolume metric quantifies the optimality and diversity of individuals. A higher hypervolume indicates that individuals are closer to the Pareto-optimal front and more diverse in the objective space. As shown in Figure 3(a), SC-iPaaS rapidly increases its hypervolume measure in the first 30 generations and converges around the 250th generation. Figure 3(a) also compares SC-iPaaS with NSGA-II, which is a well-known evolutionary multiobjective optimization algorithm [11]. It illustrates that SC-iPaaS yields significantly higher hypervolume than NSGA-II does.

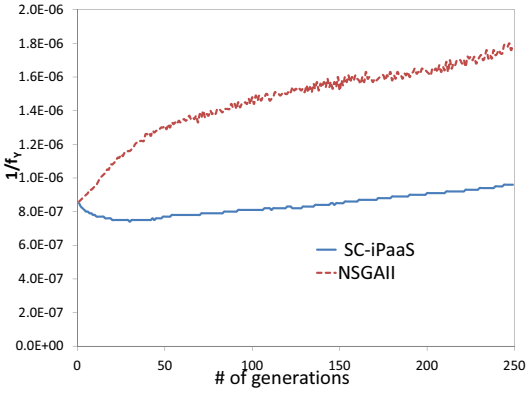
Figure 3(b) shows how hypervolume measure changes when constraints are specified for data yield and energy consumption: $C_Y = 800,000$ and $C_E = 60,000$. Both SC-iPaaS and NSGA-II quickly increases hypervolume measures in the first 40 generations; however, they are lower compared to the hypervolume measures in Figure 3(a) due to given constraints.



(a) Bandwidth Consumption (f_B)



(b) Energy Consumption (f_E)

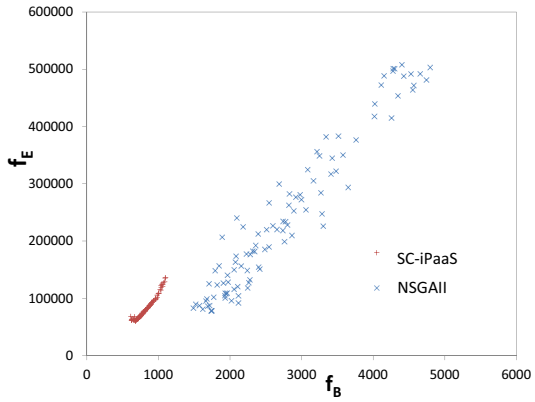


(c) Data Yield (f_Y)

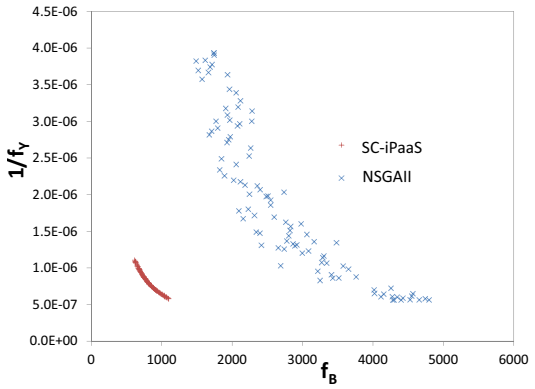
Fig. 4. Average Objective Values over Generations

While most individuals in the population are infeasible in the first 40 generations, several feasible individuals emerge around the 40th generation. Since then, infeasible individuals are gradually replaced by feasible ones. Around the 60th generation, all individuals are feasible in the population. In comparison with NSGA-II, SC-iPaaS yields a higher hypervolume value at the last generation.

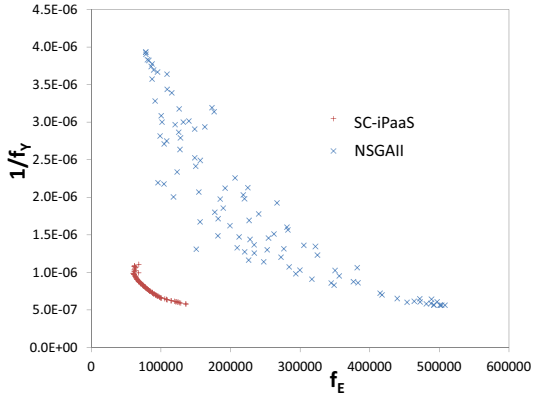
Figures 3(a) and (b) demonstrate that SC-iPaaS outperforms NSGA-II in both cases with and without optimization constraints.



(a) Bandwidth Consumption and Energy Consumption



(b) Data Yield and Bandwidth Consumption



(c) Data Yield and Energy Consumption

Fig. 5. Two-dimensional Objective Spaces

Figure 4 shows how the average objective values of individuals change over generations when no constraints are specified: $C_Y = 0$ and $C_E = \infty$. As the number of generations grow, the average bandwidth consumption and energy consumption decreases while data yield increases. Figures 3 and 4 verify that SC-iPaaS allows individuals to efficiently evolve and improve their quality and diversity.

Figure 5 shows two-dimensional objective spaces that plot individuals obtained at the last generation. (No constraints are specified.) Figure 5(a) illustrates that bandwidth consumption and energy consumption objectives correlate. As bandwidth consumption grows, energy consumption increases. Figure 5(b) illustrates that the data yield and bandwidth consumption objectives conflict with each other. As bandwidth consumption grows, data yield increases. According to Figure 5(c), the data yield and energy consumption objectives conflict with each other. As data yield increases, energy consumption increases. SC-iPaaS successfully reveals the relationships among optimization objectives and clearly exhibits the trade-offs among individuals in the population. Moreover, Figure 5 demonstrates that the individuals of SC-iPaaS outperform those of NSGA-II.

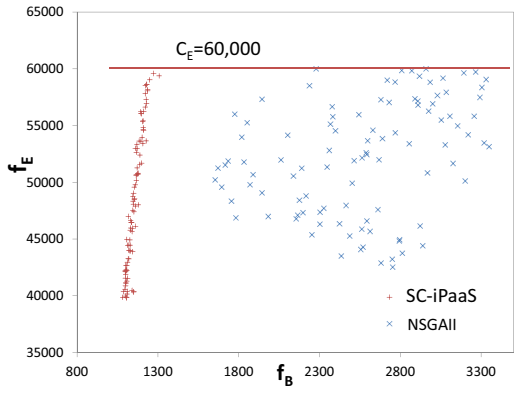
Figure 6 shows two-dimensional objective spaces that plot individuals obtained at the last generation when constraints are specified. At the last generation, all individuals are feasible and their objective values are below constraint values. In comparison with Figure 5, Figure 6 demonstrates that SC-iPaaS successfully performs its constraint handling mechanism described in Section IV-C and effectively seek Pareto-optimal data transmission configurations subject to given constraints.

VI. Related Work

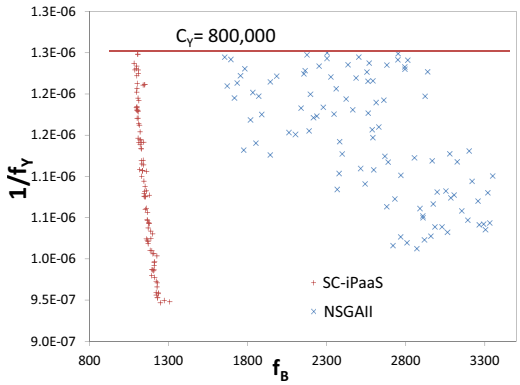
Various architectures and research tools have been proposed for cloud-integrated sensor networks [4], [13]–[20]. Hassan et al. [13], Aberer et al. [15], Shneidman et al. [16], [17] and Boonma et al. [4], [18] assume three-tier architectures similar to SC-iPaaS and investigate publish/subscribe communication between the edge layer to the cloud layer. Their focus is placed on push communication. In contrast, SC-iPaaS investigates push-pull hybrid communication between the sensor layer and the cloud layer through the edge layer. Fortino et al. study a three-tier architecture to integrate body area networks with clouds [19], [20]. Yuriyama et al. propose a two-tier architecture that consists of the sensor and cloud layers [14]. The architectures proposed by Fortino et al. and Yuriyama et al. are similar to SC-iPaaS in that they leverage the notion of virtual sensors. However, they do not consider push-pull (nor publish/subscribe) communication. All the above-mentioned work do not consider communication optimization as SC-iPaaS does.

Push-pull hybrid communication has been studied well in sensor networks [21]–[26]. However, few attempts have been made to investigate it between the edge and cloud layers in the context of cloud-integrated sensor networks. Unlike existing relevant work, this paper formulates an optimization problem with cloud-specific optimization objectives as well as the ones in sensor networks and examine sensor-to-cloud communication optimization.

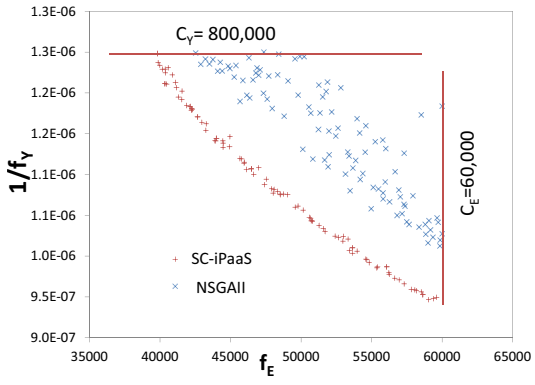
Xu et al. propose a three-tier architecture called CEB (Cloud, Edge and Beneath), which is similar to SC-iPaaS, and optimize data transmission rates between layers [27]. CEB runs two optimization algorithms collaboratively: OPT-1 and OPT-2, which optimize data transmission rates between the cloud and edge layers and between the edge and sensor layers, respectively. Optimization is carried out on a sensor node by sensor node basis with respect to a single objective: energy consumption of sensor nodes. In contrast, SC-iPaaS runs a single optimization algorithm for the entire group of sensor nodes and sink nodes simultaneously with respect to multiple conflicting objectives. SC-iPaaS assumes sensor data requests with time windows to heterogeneous sensor networks while



(a) Bandwidth Consumption and Energy Consumption



(b) Data Yield and Bandwidth Consumption



(c) Data Yield and Energy Consumption

Fig. 6. Two-dimensional Objective Spaces with Constraints Enabled

CEB assumes requests without time windows to homogeneous networks.

VII. Conclusion

This paper proposes a cloud-integrated body sensor networking architecture, called SC-iPaaS, which hosts virtualized sensors in clouds and operates physical sensors through their virtual counterparts. SC-iPaaS performs push-pull hybrid communication between three layers: cloud, edge and sensor layers. This paper formulates an optimization problem for SC-iPaaS to seek the optimal data transmission configurations and approaches the problem with an evolutionary multiobjective optimization algorithm (EMOA). SC-iPaaS successfully optimizes data transmission configurations with respect to multiple objectives (data yield, bandwidth consumption and energy consumption) subject to given constraints. It also reveals the relationships among objectives and clearly exhibits the trade-offs among different data transmission configurations. It is verified that the proposed optimizer in SC-iPaaS outperforms a well-known existing EMOA.

References

[1] C. Chen, A. Knoll, H.-E. Wichmann, and A. Horsch, "Integrating wireless sensor networks with the grid," *Journal of Modern Internet of Things*, vol. 2, no. 3, pp. 24–34, 2013.

[2] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, "A review of wearable sensors and systems with application in rehabilitation," *Journal of Neuroengineering and Rehabilitation*, vol. 9, no. 21, 2012.

[3] Y. Hao and R. Foster, "Wireless body sensor networks for health-monitoring applications," *Physiological Measurement*, vol. 29, no. 11, pp. R27–56, 2008.

[4] P. Boonma and J. Suzuki, "TinyDDS: An interoperable and configurable publish/subscribe middleware for wireless sensor networks," in *Principles and Apps. of Dist. Event-Based Systems*, A. Hinze and A. Buchmann, Eds. IGI Global, 2010, ch. 9.

[5] M. P. Hansen and A. Jaszkiewicz, "Evaluating the quality of approximations of the non-dominated set," Technical University of Denmark, Tech. Rep. IMM-REP-1998-7, 1998.

[6] E. Zitzler, J. Knowles, and L. Thiele, "Quality assessment of pareto set approximations," in *Multiobjective Optimization: Interactive and Evolutionary Approaches*, J. Branke, K. Deb, K. Miettinen, and R. Slowinski, Eds. Springer, 2008.

[7] D. Brockhoff, T. Wagner, and H. Trautmann, "On the properties of the R2 indicator," in *Proc. ACM Int'l Genetic and Evol. Computat. Conference*, 2012.

[8] E. Zitzler, L. Thiele, and J. Bader, "SPAM: Set preference algorithm for multiobjective optimization," in *Proc. of Int'l Conference on Parallel Problem Solving From Nature*, 2008.

[9] H. Trautmann, T. Wagner, and D. Brockhoff, "R2-EMOA: Focused multiobjective search using R2-indicator-based selection," in *Proc. Learning and Intelligent Optimization Conference*, Jan. 2013.

[10] K. Deb, K. Sindhya, and T. Okabe, "Self-adaptive simulated binary crossover for real-parameter optimization," in *Proc. of ACM Int'l Genetic and Evolutionary Computation Conference*, 2007.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans Evol. Computat.*, vol. 6, no. 2, 2002.

[12] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms: A comparative study," in *Proc. Int'l Conf. on Parallel Problem Solving from Nature*, 1998.

[13] M. M. Hassan, B. Song, and E.-N. Huh, "A framework of sensor-cloud integration opportunities and challenges," in *Proc. the 3rd ACM Int'l Conference on Ubiquitous Info. Mgt. and Comm.*, 2009.

[14] M. Yuriyama and T. Kushida, "Sensor-cloud infrastructure - physical sensor management with virtualized sensors on cloud computing," in *Proc. the 13th Int'l Conf. on Network-Based Info. Sys.*, 2010.

[15] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks," in *Proc. the 8th IEEE Int'l Conference on Mobile Data Management*, 2007.

[16] M. Gaynor, M. Welsh, S. Moulton, A. Rowan, E. LaCombe, and J. Wynne, "Integrating wireless sensor networks with the grid," *IEEE Internet Computing*, July/August 2004.

[17] J. Shneidman, P. Pietzuch, J. Ledlie, M. Roussopoulos, M. Seltzer, and M. Welsh, "Hourglass: An infrastructure for connecting sensor networks and applications," Harvard University, TR-21-04, Tech. Rep., 2004.

[18] P. Boonma and J. Suzuki, "Toward interoperable publish/subscribe communication between wireless sensor networks and access networks," in *Proc. IEEE Int'l Workshop on Information Retrieval in Sensor Networks*, 2009.

[19] G. Fortino, D. Parisi, V. Pirrone, and G. D. Fatta, "BodyCloud: A SaaS approach for community body sensor networks," *Future Generation Computer Systems*, vol. 35, no. 6, pp. 62–79, 2014.

[20] G. Fortino, M. Pathan, and G. D. Fatta, "BodyCloud: Integration of cloud computing and body sensor networks," in *Proc. 4th IEEE Int'l Conference on Cloud Computing Technology and Science*, 2012.

[21] H. Wada, P. Boonma, and J. Suzuki, "Chronus: A spatiotemporal macroprogramming language for autonomic wireless sensor networks," in *Autonomic Network Mgt. Principles: From Concepts to Applications*, N. Agoulmine, Ed. Elsevier, 2010, ch. 8.

[22] P. Boonma, Q. Han, and J. Suzuki, "Leveraging biologically-inspired mobile agents supporting composite needs of reliability and timeliness in sensor applications," in *Proc. IEEE Int'l Conf. on Frontiers in the Convergence of Biosci. and Info. Tech.*, 2007.

[23] W. Liu, Y. Zhang, W. Lou, and Y. Fang, "Managing wireless sensor networks with supply chain strategy," in *Proc. Int'l Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*, 2004.

[24] W. C. Lee, M. Wu, J. Xu, and X. Tang, "Monitoring top-k query in wireless sensor networks," in *Proc. IEEE International Conference on Data Engineering*, 2006.

[25] S. Kapadia and B. Krishnamachari, "Comparative analysis of push-pull query strategies for wireless sensor networks," in *Proc. International Conference on Distributed Computing in Sensor Systems*, 2006.

[26] M. Li, D. Ganesan, and P. Shenoy, "PRESTO: Feedback-driven data management in sensor networks," in *Proc. USENIX Symposium on Networked Systems Design and Implementation*, 2006.

[27] Y. Xu, S. Helal, M. Thai, and M. Scmalz, "Optimizing push/pull envelopes for energy-efficient cloud-sensor systems," in *Proc. the 14th ACM Int'l Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2011.