# Joint Traffic Engineering and Sensor Selection in Green Software Defined Internet of Things


Chih-Hang Wang†, Jian-Jhih Kuo♯, De-Nian Yang♯, and Wen-Tsuen Chen†♯

†Department of Computer Science, National Tsing Hua University, Hsin-Chu 300, Taiwan

♯Institute of Information Science, Academia Sinica, Nankang, Taipei 115, Taiwan

E-mail: s100062591@m100.nthu.edu.tw, and {lajacky, dnyang, chenwt}@iis.sinica.edu.tw



***Abstract*—123**



***Keywords*—IEEEtran, journal, LaTeX, paper, template.**


## I. Introduction

The Internet of Things (IoT) has been recognized as one of the most important network paradigm in near future, in which the devices equipped with sensors are capable of sensing real-time information from the environment. In stead of involving only a few devices, an IoT service, such as environmental monitoring, usually involves a large number of devices, and therefore overwhelms the network with such a big data. To overcome this problem, several network paradigms, such as mobile cloud computing (MCC) [] and mobile edge computing (MEC) [], emerged to alleviate the load of networks in the aspect of both traffic offloading or computation offloading.

In the past few years, MEC has grasped major attention in the mobile computing domain since it provides much lower latencies and jitters compared to the traditional cloud computing [][][]. Accordingly, the MEC brings many advantages to IoT applications that require local and real-time computation. Moreover, the big IoT data generated from the distributed IoT devices

## II. Related Work

## III. System Model and Problem Formulation

In this section, we first introduce the system model and then formulate the Traffic Engineering and Sensor Selection Problem (TES$^2$P). We also prove that TES$^2$P is NP-hard and inapproximable within $\ln n$ where $n$ is the number of observed locations.

### A. Network Model

We consider an SD-IoT system which consists of wireless sensor network (WSN) and software defined network (SDN) with mobile edge computing (MEC) paradigm as shown in Fig. 1. The WSN consists of $N$ sensors $\mathbb{N} = \{1, 2, ..., N\}$ and $K$ observed locations $\mathbb{K} = \{1, 2, ..., K\}$. Each sensor $n \in \mathbb{N}$ with a set of covering locations $C_n \subseteq \mathbb{K}$ associates to a base station (BS) and detects the status of each location $k \in C_n$ with a success probability $\alpha_{n,k}$ [1]. Let $b(n)$ be the data rate of sensor $n$. For each observed location $k \in \mathbb{K}$, we assume that a location is successfully detected if the overall success probability on $k$ is greater than the threshold $I (I \leq 1)$. If the location is successfully detected by some sensors, the data of

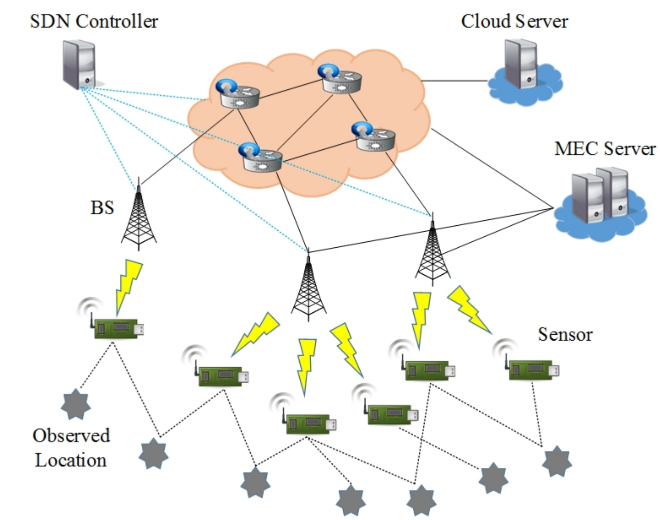



Fig. 1. SD-IoT system

other sensors that is also enable to detect this location can be reconstructed due to the spatial correlation of data [2]. Note that we only consider the success probability of detection and assume that the sensory data is always successfully transmitted to its serving BS.

Let $G = (V, E)$ be a SDN with MEC, where $V$ is the set of nodes, which includes a cloud server $D$, $R$ MEC servers $\mathbb{R} = \{1, 2, ..., R\}$, and multiple SDN-enabled BSs and switches, and $E$ is the set of links between the nodes in $V$. For each link $l \in E$, let $B(l)$ denote the capacity of $l$ which limits the total rate of transmitted flows on $l$. The cloud server $D$ takes responsibility to collect the sensory data and reconstruct the data of unselected sensors, while each MEC server $r \in \mathbb{R}$ takes charge of aggregating multiple flows into a single one and compressing them with a ratio $\beta$. Each MEC server $r$ may associate to some BSs and switches. In terms of flexibility, each flow can either be directly transmitted to the cloud server or aggregated with other flows on a MEC server to reduce the load of SDN as well as the energy consumption. Note that each flow can be aggregated and compressed with other flows at most once.

### B. Energy Consumption Model

In this paper, we consider the energy consumption of SD-IoT in the following three folds: 1) the cloud server, 2) the MEC server, and 3) the switch. For the cloud server, we consider the energy consumption for reconstructing unknown data which is proportional to the number of sensory data (e.g., compressive sensing [2], [3]). Therefore, the energy consumption model of cloud server can be formulated as follows.

$$\delta = \sum_{n \in \mathbb{N}} x_n \cdot \mathcal{E}(\mathcal{P}) \tag{1}$$

where $x_n$ denotes if sensor $n$ is selected and $\mathcal{E}(\mathcal{P})$ is the energy function of data reconstruction policy $\mathcal{P}$.

For the MEC server, the energy consumption is proportional to the computational load of CPU. In other words, if more flows are aggregated on a MEC server, more energy is consumed. Therefore, according to [4], the energy consumption of MEC server $v$ can be modeled as follows.

$$\sigma_v = e_{idle} + e_{unit} \cdot \sum_{n \in \mathbb{N}} \sum_{p \in P_{s_n,v}} \pi_{f_n,s_n,v,p} \tag{2}$$

where $e_{idle}$ is the idle energy consumption, $e_{unit}$ is the unit energy consumption to aggregate one flow, $\pi_{f_n,s_n,v,p}$ denotes if the sensor flow $f_n$ originated from BS $s_n$ is aggregated on $v$ on the routing path $p$, and $P_{s_n,v}$ is the set of path from $s_n$ to $v$ (detailed later).

According to [5], the energy consumption of switch to forward flows is a linear function of flow rate which can be modeled as follows.

$$\mathcal{F}(b(n)) = e_{config} \cdot \frac{b(n)}{config} \tag{3}$$

where $config$ is a constant and can be 10Mbps, 100Mbps, or 1Gbps and $e_{config}$ is the power for a port running at line rate $config$. Note that we omit the constant part in this model since it does no effect on the optimization problem. In addition, we assume that the consumed energy function of BS and switch to forward flows are identical. To simplify the presentation, unless otherwise specified, both the terms "switch" and "BS" are called "switch" when we calculate the consumed energy for data forwarding in the following of paper.

### C. Problem Formulation

In this paper, our goal is to minimize the energy consumption of SD-IoT system by jointly selecting sensors and routing the corresponding flows, while the data of unselected sensors can be reconstructed. Our problem is formulated as follows.

**Problem:** The Traffic Engineering and Sensor Selection Problem (TES$^2$P)

**Instance:** A set of sensors $\mathbb{N} = \{1, 2, ..., N\}$ with $N$ coverage sets $\mathbb{C} = \{C_1, C_2, ..., C_N\}$, a set of observed locations $\mathbb{K} = \{1, 2, ..., K\}$, and a SDN $G = (V, E)$.

**Task:** To jointly select sensors and determine their paths such that the total energy consumption of SD-IoT system is minimized, where the energy consumption includes that of the cloud server, MEC servers, and switches.

In the following, we propose the mathematical formulation for TES$^2$P. Our formulation contains the following decision variables. Binary variable $x_n$ denotes if sensor $n$ is selected. Since each flow is allowed to be directly transmitted to the cloud server or aggregated with other flows on a MEC server, we let binary variable $\pi_{f_n,s_n,v,p}$ denote if the sensor flow $f_n$ originated from BS $s_n$ is aggregated on $v$ on the routing path $p$, where $v \in R \cup \{D\}$, $p \in P_{s_n,v}$, and $P_{s_n,v}$ is the set of paths from $s_n$ to $v$. Note that if a flow is aggregated on the cloud server (i.e., $v = D$), it means that the flow is directly routed to the cloud server without any aggregation on a MEC server. For each aggregation flow $f_v$, we let binary variable $\pi^{aggr}_{f_v,v,D,p}$ indicate if $f_v$ traverses path $p$ from $v$ to $D$, where $v \in R$, $p \in P_{v,D}$.

Recall that we consider the energy consumption in three folds: 1) the cloud server, 2) the MEC server, and 3) the switch. Thereby, the objective function is as follows.

$$\begin{aligned} min \quad & \delta + \sum_{v \in \mathbb{R}} \sigma_v \\ & + \sum_{n \in \mathbb{N}} \sum_{v \in \mathbb{R} \cup \{D\}} \sum_{p \in P_{s_n,v}} \sum_{u \in p} \pi_{f_n,s_n,v,p} \cdot \mathcal{F}(b(n)) \\ & + \sum_{v \in \mathbb{R}} \sum_{p \in P_{v,D}} \sum_{u \in p} \pi^{aggr}_{f_v,v,D,p} \cdot \mathcal{F}(b^{aggr}(f_v)) \end{aligned}$$

where $u$ is the index of switch on the path $p$ and $b^{aggr}(f_v)$ is the data rate of aggregation flow $f_v$. The objective function minimizes the total energy consumption of cloud server, MEC servers, and switches. The last two terms are the energy consumption of switches for forwarding non-aggregated and aggregated flows, respectively. The problem has the following constraints:

$$1 - \prod_{n | k \in C_n} (1 - x_n \cdot \alpha_{n,k}) \geq I, \quad \forall k \in \mathbb{K} \tag{4}$$

$$\sum_{v \in \mathbb{R} \bigcup \{D\}} \sum_{p \in P_{s_n,v}} \pi_{f_n,s_n,v,p} \geq x_n, \quad \forall n \in \mathbb{N} \tag{5}$$

$$\sum_{p \in P_{s_n,v}} \pi_{f_n,s_n,v,p} \leq \sum_{p \in P_{v,D}} \pi^{aggr}_{f_v,v,D,p}, \quad \forall n \in \mathbb{N}, v \in \mathbb{R} \tag{6}$$

$$b^{aggr}(f_v) = \sum_{n \in \mathbb{N}} \sum_{p \in P_{s_n,v}} \pi_{f_n,s_n,v} \cdot b(n) \cdot \beta \quad \forall v \in \mathbb{R} \tag{7}$$

$$\begin{aligned} & \sum_{n \in \mathbb{N}} \sum_{v \in \mathbb{R} \bigcup \{D\}} \sum_{p \in P_{s_n,v} | l \in p} \pi_{f_n,s_n,v,p} \cdot b(n) \\ & + \sum_{p \in \mathbb{R}} \sum_{p \in P_{v,D} | l \in p} \pi^{aggr}_{f_v,v,D,p} \cdot b^{aggr}(f_v) \leq B(l), \quad \forall l \in E \end{aligned} \tag{8}$$

$$x_n \in \{0,1\}, \pi^{aggr}_{f_v,v,D,p} \in \{0,1\}, \quad \forall n \in \mathbb{N}, v \in \mathbb{R}, p \in P_{v,D} \tag{9}$$

$$\pi_{f_n,s_n,v,p} \in \{0,1\}, \quad \forall n \in \mathbb{N}, v \in \mathbb{R} \cup \{D\}, p \in P_{s_n,v} \tag{10}$$

Constraint (4) ensures that each location is observed by sufficient sensors such that the unknown data with spatial correlation can be reconstructed. Constraint (5) shows that if a sensor is selected, the corresponding flow is generated in the network. Constraint (6) states that the aggregation flow $f_v$ is generated from node $v$ if at least one sensor flow is aggregated and compressed on $v$. Constraint (7) calculates the size of aggregation flow which is the sum of rate of aggregated flows times the compression ratio $\beta$, and constraint (8) ensures that the total flow rate traversing on each link $l$ does not exceed the capacity of $l$. Finally, constraints (9) and (10) ensure that the binary variables belong to $\{0, 1\}$. In the following, we prove that the problem is NP-hard with the reduction from Weighted Set Cover Problem (WSCP).

**Theorem 1.** TES$^2$P is NP-hard.

*Proof:* We first identify the special case of TES$^2$P by defining 1) the success probability of each sensor on each location in its coverage range is 1, 2) the energy consumption of cloud server and MEC servers are zero, 3) the link capacity $B(l)$ of each link $l$ is infinite, and 4) the compression ratio $\beta$ is 1. Therefore, the problem becomes selecting a set of sensors to cover all the locations such that the energy consumption to forward the sensor flows is minimized. In the following, we prove this special case is NP-hard with the reduction from WSCP.

Let $\mathbb{U}$ denote the set of elements. Let $\mathbb{S} = \{S_1, S_2, ..., S_m\}$ be the collection of sets in WSCP such that each set $S_i$ in $\mathbb{S}$ with weight $w_i$ is a subset of $\mathbb{U}$, and $\bigcup_{S_i \in \mathbb{S}} \{S_i\} = \mathbb{U}$. The problem is to find a subset $\mathbb{S}^* \subseteq \mathbb{S}$ with $\bigcup_{S_j \in \mathbb{S}^*} \{S_j\} = \mathbb{U}$ such that the total weight $\sum_{j \in \mathbb{S}^*} w_j$ is minimized. For each set of WSCP, we construct an instance of TES$^2$P as follows. For each set $S_i$, we construct a sensor $n$ and its associated BS and the BS directly connects to the cloud server in one hop. Specifically, let $\mathbb{K} = \{1, 2, ..., K\} = \mathbb{U}$ as the set of observed locations, and $\mathbb{N} = \{1, 2, ..., N\} = \mathbb{S}$ as the set of sensors. We denote $C_n$ as the set of locations covered by sensor $n$, where $\bigcup_n C_n = \mathbb{K}$. In the reduction, we set the energy consumption of BS to forward the corresponding flow of sensor $n$ as $w_n$.

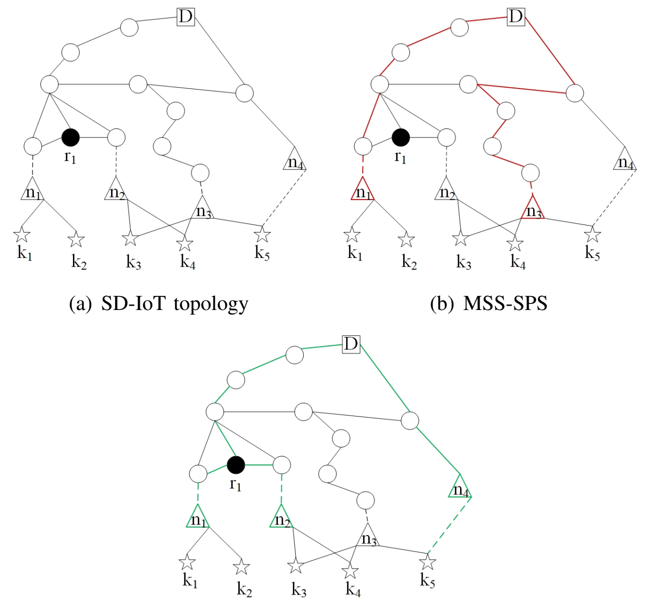
For the WSCP, if there exists a collection of set $\mathbb{S}^*$ such that the union elements in $\mathbb{S}^*$ is $\mathbb{U}$, we can find the corresponding subset $\mathbb{N}^* \subseteq \mathbb{N}$ such that the union of observed locations in $\mathbb{N}^*$ is also $\mathbb{K}$. Conversely, if there exists a subset $\mathbb{N}^* \subseteq \mathbb{N}$ with the union of covered targets in $\mathbb{N}^*$ as $\mathbb{K}$, we can find the corresponding collection of sets $\mathbb{S}^*$ such that the union in $\mathbb{S}^*$ is also $\mathbb{U}$. This is because each sensor in $\mathbb{N}$ with weight $w_n$ corresponds to a coverage set in $\mathbb{S}$ with weight $w_i$. Hence, the special case of TES$^2$P is as hard as WSCP, and therefore TES$^2$P is NP-hard. The theorem follows. ■

From Theorem 1, TES$^2$P is at least hard as WSCP. Feige [6] proved that WSCP is inapproximable within $\ln n$ and thus we have the following result.

**Corollary 1.** TES$^2$P is inapproximable within $\ln n$.

## IV. Algorithm Design

In the following subsections, we first propose a baseline algorithm, minimum sensor selection and shortest path search (MSS-SPS) algorithm, and explain why this scheme is not suitable for SD-IoT. We then devise minimum energy sensor selection (MESS) algorithm to minimize the overall energy consumption of SD-IoT, while guaranteeing the data reconstruction and aggregation requirement of SD-IoT. Note that we first focus on the data reconstruction and aggregation constraints instead of the link capacity because the big data processing in SD-IoT has not been well explored in previous research and MEC has been regarded as one of the most important technique for the next generation network. Next, we derive the approximation ratio and time complexity of MESS. Finally, we extend MESS to the scenario with limited link capacity and propose a re-routing and sensor re-selection scheme.




(a) SD-IoT topology (b) MSS-SPS

(c) MESS

Fig. 2. An illustrative example of SD-IoT system

### A. Baseline Algorithm

Since we aim to minimize the energy consumption of SD-IoT by jointly selecting sensors and routing the corresponding flows, an intuitive method is to first select the minimum number of sensors to guarantee the data reconstruction requirement and then iteratively route the sensor flows by shortest path scheme to minimize the forwarding energy of switches. MSS-SPS first iteratively selects the sensor that covers the most amount of unsatisfied locations[1] until all the locations are satisfied. Then, MSS-SPS sequentially routes the sensor flows by adopting the shortest path scheme. Afterward, MSS-SPS calculates the energy consumption of SD-IoT with the above selection and routing results by (1), (2), and (3).

**Example.** Fig. 2 shows an illustrative example with one cloud server, one MEC server, four sensors, and five locations. In this example, we assume that the compression ratio is 0.5, and the unit energy consumption of 1) cloud server, 2) MEC server, and 3) switch/BS are 2, 2, and 10, respectively. Moreover, we assume that each flow is a unit flow and the success probability of each sensor on each location in its coverage set is 1. MSS-SPS first selects sensor $n_3$ because it covers the most amount of unsatisfied locations and then it selects $n_1$ since it covers the remaining two locations. Next, MSS-SPS routes $n_1$ and $n_3$ by adopting shortest path scheme which results in 4 hops and 5 hops to the cloud server $D$ for $n_1$ and $n_3$, respectively. Therefore, on BSs and switches, $n_1$ leads to $4 \times 10 = 40$ energy consumption, while $n_3$ results in $5 \times 10 = 50$ energy consumption. Since both $n_1$ and $n_3$ are directly transmitted to $D$, the energy consumption of MEC server is zero. For the energy consumption of cloud server, the two selected sensors result in $2 \times 2 = 4$. Therefore, the total energy consumption is $40+50+4 = 94$. Fig. 2(b) shows the result of MSS-SPS, where the selected sensors and their corresponding paths to the cloud

[1] In this paper, we call a location is *unsatisfied* if its success probability has not been satisfied by the selected sensors; otherwise, it is called *satisfied*.

server are marked in red. MSS-SPS results in excess consumed energy in SD-IoT since because the aggregation benefit and the tradeoff between energy consumption of data reconstruction and routing are not carefully exploited.

### *B. Approximation Algorithm*

To solve TES$^2$P, we propose minimum energy sensor selection (MESS) algorithm, which jointly selects the sensors and determines the path of sensor flows to cover the locations such that the total energy consumption of SD-IoT is minimized. MESS first chooses the path to the cloud server for each sensor and sets their energy cost. Then, it iteratively selects the sensor with the minimum cost-performance ratio until the required success probability on all locations are satisfied. The cost-performance ratio of sensor $n$ is defined as follows.

$$CP_n = \frac{\phi_n}{\sum_{k \in C_n \cap \mathbb{Y}} min\{I - \theta_k, \theta_{k,n} - \theta_k\}}, \quad (11)$$

where $\mathbb{Y}$ is the set of locations that has not been satisfied, $\phi_n$ is the energy cost of sensor $n$, $\theta_k$ is the current success probability of location $k$, and $\theta_{k,n}$ is the success probability of location $k$ after sensor $n$ is selected; that is, $\theta_{k,n} = 1 - (1 - \alpha_{n,k}) \prod_{n' \in \mathbb{Z} | k \in C_{n'}} (1 - x_{n'} \cdot \alpha_{n',k})$, where $\mathbb{Z}$ is the set of sensors that has been selected previously. Overall, MESS consists of two phases: 1) path and cost decision (PCD) and 2) greedy selection (GS). The first phase determines the path with the minimal energy consumption for each sensor, and the second phase iteratively selects the sensor with the minimum $CP$ to minimize the energy consumption of SD-IoT while satisfying the data reconstruction requirement.

#### *1) Path and Cost Decision (PCD)*

In this phase, MESS determines the path with the minimum energy consumption for each sensor $n \in \mathbb{N}$ and set the consumed energy on the path as $\phi_n$. Recall that we consider the energy consumption of 1) cloud server, 2) MEC server, and 3) switch, and therefore the energy consumption of a path includes the above three terms. PCD considers the energy consumption of both aggregation and non-aggregation paths for each sensor.

Since the flow size $b(n)$ is not changed on the non-aggregation path to the cloud server $D$, MESS finds the shortest path from $s_n$ to $D$ and calculates the energy dissipation as the product of hop counts of the path and the consumed energy for forwarding the flow by (3). Moreover, for data reconstruction on the cloud server, PCD calculates the energy consumption $\mathcal{E}(\mathcal{P})$ according to (1). Therefore, the energy consumption of non-aggregation path is calculated as $\mathcal{F}(b(n)) \cdot \sum_{u \in P^{st}_{s_n,D}} 1 + \mathcal{E}(\mathcal{P})$, where $P^{st}_{s_n,D}$ denotes the shortest path from $s_n$ to $D$.

For the aggregation path of flow $f_n$, since the flow size becomes smaller after being aggregated and compressed on a MEC server, PCD first calculates the energy consumption of aggregation path from the serving BS $s_n$ to the cloud server $D$ by computing the energy consumption of paths from $s_n$ to the MEC server $v \in \mathbb{R}$ and from $v$ to $D$. More specifically, for each $v$, PCD calculates the energy consumption of the path from $s_n$ to $D$ via $v$ as $\mathcal{F}(b(n)) \cdot \sum_{u \in P^{st}_{s_n,v}} 1 + \mathcal{F}(b(n) \cdot \beta) \cdot \sum_{u \in P^{st}_{v,D}} 1$, where $P^{st}_{s_n,v}$ and $P^{st}_{v,D}$ are the shortest path from $s_n$ to $v$ and $v$ to $D$, respectively. In addition to the energy consumption of cloud server for data reconstruction, PCD further calculates the energy consumption of MEC server for aggregation by (2). That is, each sensor $n$ causes $e_{unit}$ consumed energy to the MEC server if it is aggregated on the server. Therefore, the total energy consumption of aggregation path from $s_n$ to $D$ via $v$ is calculated as $\mathcal{F}(b(n)) \cdot \sum_{u \in P^{st}_{s_n,v}} 1 + \mathcal{F}(b(n) \cdot \beta) \cdot \sum_{u \in P^{st}_{v,D}} 1 + \mathcal{E}(\mathcal{P}) + e_{unit}$, and PCD calculates for each $v$.

After getting the energy consumption of both aggregation and non-aggregation paths, PCD selects the path with the minimum energy consumption for $n$ and sets its energy cost $\phi_n$ to the energy consumption value. PCD runs until every sensor is assigned a path and set energy cost. Then, MESS goes to GS.

**Example.** Again, we consider the illustrative example shown in Fig. 2(a). To simplify the presentation, we only explain the detail behavior for few sensors and the remaining process are similar. For sensor $n_1$, PCD firsts calculates the energy consumption of its aggregation and non-aggregation path. The energy consumption of non-aggregation path is calculated as $4 \times 10 + 2 = 42$ because the shortest from the serving BS of $n_1$ to $D$ consists of 4 hops and $n_1$ causes 2 consumed energy to the cloud server for data reconstruction. On the other hand, the consumed energy on the aggregation path is only $10 + 4 \times 5 + 2 + 2 = 34$ since the flow size becomes half (we assume $\beta = 0.5$) after traversing the MEC server $r_1$, and therefore the consumed energy to forward it becomes half (i.e., 5) and the flow causes 2 consumed energy to both cloud server and MEC server. Therefore, PCD selects the aggregation path for $n_1$ since it consumes less energy and set the energy cost of $n_1$ to 34. The final result in this phase is that $n_1$ and $n_2$ select their aggregation paths, while $n_3$ and $n_4$ select their non-aggregation paths since these paths consumes the minimum energy. Moreover, the energy cost of $n_1$, $n_2$, $n_3$, and $n_4$ are 34, 34, 52, and 12, respectively.

#### *2) Greedy Selection (GS)*

In this phase, MESS iteratively selects the sensor with the minimum CP until all the locations are satisfied. First, GS calculates CP for each sensor as defined in (11). Then, GS selects the sensor with the minimum CP, that is, the selected sensor exerts a lower energy cost for an unsatisfied location. Next, GS checks whether the locations in the coverage set of the selected sensor are satisfied. If the location $k$ is satisfied, that is, $1 - \prod_{n' \in \mathbb{Z} | k \in C_{n'}} (1 - x_{n'} \cdot \alpha_{n',k}) \geq I$, GS moves it out from the unsatisfied location set $\mathbb{Y}$; otherwise, GS updates $\theta_k$ as $1 - \prod_{n' \in \mathbb{Z} | k \in C_{n'}} (1 - x_{n'} \cdot \alpha_{n',k})$. Afterward, GS goes to the next iteration and stops until all the locations are satisfied.

**Example.** Recall that the energy cost of $n_1$, $n_2$, $n_3$, and $n_4$ are 34, 34, 52, and 12, respectively after the PCD phase. GS first calculates the cost performance ratio of $n_1$, $n_2$, $n_3$, and $n_4$ and they are 17, 17, $\frac{52}{3}$, and 12, respectively. Therefore, GS selects $n_4$ in the first iteration and removes $k_5$ out from the unsatisfied location set. In this example, note that we assume the success probability of each sensor on each location in its coverage set is 1, and therefore the location must be satisfied by selecting one of the sensors covering it. Afterward, GS selects $n_1$ in the next iteration since it has the minimum cost performance ratio among the remaining sensors. Finally, GS selects $n_2$ and the total energy consumption is $34 + 34 + 12 =$

80, while the baseline algorithm MSS-SPS results in 94. Fig. 2(c) shows the result of MESS, where the selected sensors and their corresponding paths to the cloud server are marked in green.

**Time complexity.** In the phase PCD, MESS first determines the path with the minimum energy consumption for each sensor $n \in \mathbb{N}$ by considering at most $R+1$ paths, which includes $R$ aggregation path and one aggregation path. The single source shortest path can be found in $\mathcal{O}(|V|^2)$ by the well-known Dijkstra's algorithm, while the energy consumption of path is calculated in a constant time. Therefore, it takes at most $\mathcal{O}(NR|V|^2)$ to find $R+1$ shortest path via different MEC servers for $N$ sensors. Then, MESS takes $\mathcal{O}(N)$ to set the energy cost. Afterward, it takes $\mathcal{O}(NR|V|^2)$ in the phase PCD.

In the phase GS, MESS iteratively selects the sensor with the minimum CP until all the locations are satisfied. It takes $\mathcal{O}(K)$ iterations to guarantee the coverage since there are at most $K$ sensors. Moreover, the minimum CP can be found in $\mathcal{O}(\log K)$ by using a priority queue in each iteration. Therefore, the time complexity of GS is $\mathcal{O}(K \log K)$ and the overall time complexity of MESS is $\mathcal{O}(NR|V|^2) + \mathcal{O}(K \log K)$.

### *C. Approximation Ratio*

To derive the approximation ratio of the proposed greedy algorithm, we first prove that the constraint is sub-modular and non-decreasing (see Definition 1), and then with the similar arguments in [], the greedy algorithm can guarantee that the solution is no more than $O(\log \gamma) \cdot OPT$, where $\gamma = \theta_t/\theta_0$. **[all the symbols can be modified later]**

**Definition 1.** Given a set $N$, a non-negative function $z : S \subseteq N \rightarrow R^+ \cup \{0\}$ is sub-modular and non-decreasing if $\forall S$, $T \subseteq N, S \subseteq T$, $(z(S \cup \{n\}) - z(S)) \geq z(T \cup \{n\}) - z(T) \geq 0$, where $n \in N$.

non-decreasing: $z(S) \leq z(T)$.

sub-modular: $z(S) + z(T) \geq z(S \cup T) + z(S \cap T)$.

### *D. Discussion*

## Acknowledgment

The authors would like to thank...

## References


[1] E. Onur, C. Ersoy, and H. Deliç, "How many sensors for an acceptable breach detection probability?" *Computer Communication*, vol. 29, no. 2, pp. 173–182, Jan. 2006.

[2] L. Kong, M. Xia, X. Y. Liu, M. Y. Wu, and X. Liu, "Data loss and reconstruction in sensor networks," in *2013 Proceedings IEEE INFOCOM*, April 2013, pp. 1654–1662.

[3] C. Karakus, A. C. Gurbuz, and B. Tavli, "Analysis of energy efficiency of compressive sensing in wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1999–2008, May 2013.

[4] C. Mbius, W. Dargie, and A. Schill, "Power consumption estimation models for processors, virtual machines, and servers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 6, pp. 1600–1614, June 2014.

[5] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A power benchmarking framework for network devices," in *Proceedings of the 8th International IFIP-TC 6 Networking Conference*, 2009, pp. 795–808.

[6] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, Jul. 1998.