IoT Communication Sharing: Scenarios, Algorithms and Implementation

Chuang Hu*, Wei Bao†, Dan Wang*
*Department of Computing, The Hong Kong Polytechnic University
†School of Information Technology, The University of Sydney

***Abstract*—Nowadays, manufacturers want to collect the data of their sold-products to the cloud, so that they can conduct analysis and improve the operation, maintenance and services of their products. Manufacturers are looking for a self-contained solution for data transmission since their products are typically deployed in a large number of different buildings, and it is neither feasible to negotiate with each building to use the building's network (e.g., WiFi) nor practical to establish its own network infrastructure. ISPs are aware of this market. Since the readily available 3G/4G is over costly for most IoT devices, ISPs are developing new choices. Nevertheless, it can be expected that the choices from ISPs will not be fine-grained enough to match hundreds or thousands of requirements on different costs and data volumes from the IoT applications. To address this problem, we for the first time propose IoT communication sharing (ICS). We first clarify the ICS scenarios. We then formulate the IoT communication sharing (ICS) problem, and develop a set of algorithms with provable performance. We further present our implementation of a fully functioning system. Our evaluations show that ICS and our algorithms can lead to a cost reduction of five times and eight times respectively for the two real-world cases.**

I. Introduction

One important value proposition of the Internet of Things (IoT) is the data generated by the IoT devices (a.k.a, things) [1]. When sending such data to the cloud, with state-of-the-art data mining techniques and the computational power of the cloud, the adding value can be significant [2]. For example, it has been shown that big building data (e.g., carbon dioxide (CO2) data from the heating, ventilation and air conditioning (HVAC) systems) can be exploited to predict traffic status of nearby roads [3]. Smart After-sales Maintenance and Services (SAMS), which will become the case study of this paper, is another example. Manufacturers of air conditioners, pumps, elevators, etc., are now transforming their machinery into smart machinery. When sending the data of their products to the cloud, SAMS can operate in a trouble-preventing mode instead of trouble-shooting mode. This can substantially improve the quality and reduce the cost of the product maintenance. Moreover, manufacturers can learn the usage patterns of their customers. Thus, they can recommend other products and develop top-up services based on such knowledge [4].

To fully realize the aforementioned applications, one key question remains to be answered: how to transmit the data from the things to the cloud, in an easy-to-use and cost-effective way?

The vendor may develop a WiFi network for the IoT application. However, WiFi needs additional infrastructure, e.g., a gateway that finally relays data to the cloud. This is not suitable for SAMS. For example, a vendor would like to monitor all its air conditioners in a region, installed in a large number of buildings. The WiFi choice needs the deployment of the WiFi networks on a building-by-building basis. In other words, the vendor is developing a separated network infrastructure. If using the existing WiFi networks in the buildings, there will be policy and security concerns from buildings. Looking from the buildings' perspective, a building can easily have products from tens of vendors. If each vendor wants its equipment to infiltrate the WiFi network of the building, building operators need to bear overwhelming liability. Simply-put, applications such as SAMS are looking for an infrastructure-less (or self-contained) solution.

The vendor may rely on the infrastructure of a service provider (ISP) and subscribe a dedicated wireless communication channel for each IoT device [5] to support the *thing-to-cloud communication (TCC)* links. Current choices for TCC links are very limited. The readily available 3G/4G is over-costly for the majority of IoT devices. The industry has realized this problem and is actively developing less costly choices. User Experience-Category (CAT) represents a group of technologies with much smaller data rates and thus costs [6]. CAT1 was released in 2016 and CAT0 is under deployment [7]. Nevertheless, we may expect tens of choices of communication channels with different costs and data rates, yet we will face hundreds, if not thousands, of heterogeneous requirements. In the SAMS example, the cost of CAT1 might be justifiable for a chiller, yet it may be too costly for a fan.

We see a clear gap between the possible choices of TCC links, and the number of requirements on different costs and data rates from the IoT applications. To address this issue, we propose IoT communication sharing (ICS), where a greater number of IoT devices, with heterogeneous data communication requirements, share a fewer choices of TCC links, and transmit their data to the cloud.

In this paper, we first present an analysis of an SAMS application. We clarify the IoT communication sharing scenario and model. We then study how to optimally share the IoT communications. We formulate an IoT communication sharing (ICS) problem and develop efficient algorithms. We study ICS under a particular price model, the pay-as-you-go (PAYG) model, ICS-PAYG. PAYG is usually adopted in the early stage of a new business and is the current pricing model for CAT1. We develop an approximation algorithm and a fast heuristic for ICS-PAYG. We implement a functioning system and evaluate ICS experimentally. In a larger scale, we evaluate ICS through simulations using real-world cases. Our evaluation shows that ICS and our algorithms can lead to a cost reduction of five and eight times respectively for the two real-world cases.

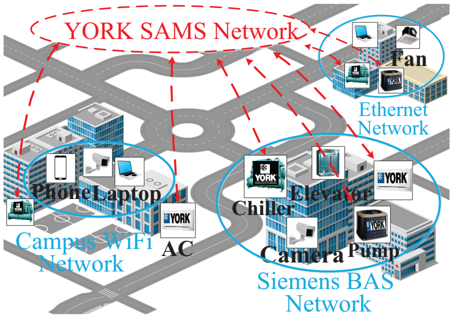

Fig. 1: The illustration of different networks.

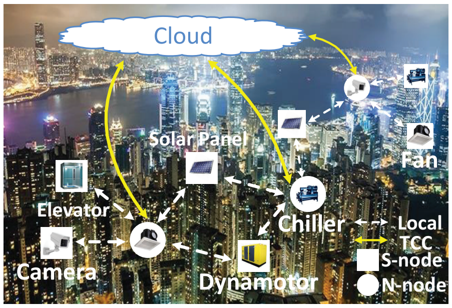

Fig. 2: Smart After-Sales Maintenance and Services (SAMS).

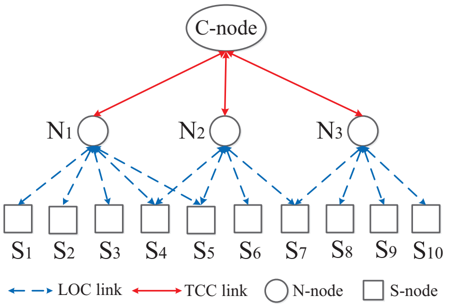

Fig. 3: The topology of Smart After-Sales Maintenance and Services.

The contributions of the paper can be summarized as:

- Using SAMS as an example application, we clarify the scenario and the model of IoT communication sharing (ICS). To the best of our knowledge, we are the first to propose and formalize the ICS model (Section II).
- We formulate the ICS optimization problem (Section III) and develop efficient algorithms (Section IV). In particular, we study ICS under the PAYG pricing model, the currently adopted model in industry (Section V).
- We implement a fully functioning system of the ICS model (Section VI) and comprehensively evaluate ICS (Section VII).

II. The SAMS Network and the IoT Communication Sharing Model

We are working on a real SAMS on centralized HVAC systems.[1] Since the SAMS applications are in the emerging states, we first analyze a concrete example in the maintenance of chillers, one core component of an HVAC system. We show that the network supporting the SAMS applications should be separated from other networks in a building. We then present the IoT communication sharing model and related studies.

A. Chiller Maintenance: How SAMS Benefits

Current chiller maintenance consists of routine maintenance and emergency repair, and their respective costs are USD $897.12 and USD $5639.94 per time [8]. An optimal maintenance plan is a balance of routine maintenance and emergency repair. This is usually done by analyzing the degradation of chillers. Intuitively, routine maintenance will be more frequent if a certain type of chiller degrades faster. Chiller degradation is affected by many factors, such as its intrinsic reliability and the usage pattern of the chiller (e.g., the Freon level depends on the intensity that the chiller is being used). Though the chiller reliability can be extensively tested in labs, the usage pattern of a chiller is determined by customers, and is difficult to know at the time that this chiller is being manufactured. This is one key reason that SAMS can become superior.

The key indicator for the performance (degradation) of a chiller is Coefficient of Performance (COP) [9]. Maintenance is needed if the COP of a chiller is below a certain threshold.[2]

To compare the current maintenance plan and SAMS, we conduct an illustrative analysis by using four year data of ten chillers in three buildings. We calculated the optimal plan for current maintenance with a routine maintenance interval of 3.1 months (details in [10]), leading to a cost of USD $4052.64 per chiller per year. For SAMS, we can collect the chiller data in real time. The cost reduces to USD $2813.66, with an average maintenance interval of 3.89 months. This leads to a 30.58% saving. Note that this is only a baseline comparison. If we consider joint maintenance of multiple equipment, a prediction of equipment degradation, and that current maintenance plan has to be conservative (e.g., shorter than 3.1 months), we can expect a much greater gain from SAMS.

B. The Networks in Buildings and the SAMS Network

There are many networks in a modern building. Building equipment is controlled by building automation systems (BAS) [11], [12]. BAS uses Ethernet to send data to a server located in the building. The target of BAS is to manage thousands of devices, from different vendors, within a building. BAS vendors include Schneider, Siemens, John Control, etc, which differ from equipment vendors in HVAC, elevator, etc.

There are WiFi or WAN networks in a building/campus, and a gateway is used to connect the Internet. The target of WiFi/WAN networks is to serve the network connection of personal computers/devices. The WiFi/WAN is controlled by building/campus owners.

The target of SAMS is to transmit the data of thousands of IoT devices, of the same vendor, spread at hundreds of buildings, to the cloud. The spread of the devices in buildings controlled by different building owners or BAS companies made it infeasible to use the BAS or WiFi network since a building-by-building based agreement is needed. Logically, the BAS network, the building WiFi network and the SAMS network, should be *different networks*, see Fig. 1.

We also want to comment that networks to support emerging home or office applications are also being developed, using wireless communications of Bluetooth, ZigBee and LoRa. The

[1] SAMS emphasizes on commercial equipment rather than home equipment. They are much more high-valued and the maintenance is vendor responsible.

[2] A low COP does not mean a direct chiller failure, yet it indicates sensible human comfort down grade and substantial energy usage inefficiency. The current threshold imposed in Country/City Anonymity is 5.7.

administration of these networks is the local home or office application owners. Again we cannot see the possibility that they are willing to unanimously grant the access and usage of their networks to all kinds of building equipment vendors.

C. The IoT Communication Sharing Model

We now present the IoT communication sharing (ICS) model that can efficiently support applications such as SAMS. We will present an implementation of this ICS model in Section VI.

ICS is a three-tier network (see Figs. 2–3). The first tier consists of *the clouds (C-node)*. The second tier consists of the *network nodes (N-node)*. The links between the C-nodes and N-nodes are the TCC links subscribed from ISPs, which incur costs.[3] The third tier consists of the *sensing nodes (S-node)*. S-nodes extracts data from the equipment. N-nodes and S-nodes form *local networks (LOC)* using free communication channels. Short-range channels include Zigbee, Bluetooth, etc, and longer-range channels include LoRa, SigFox, etc which can well connect devices across floors or even buildings [13].

Note that N-nodes and S-nodes are logical and can be merged and installed on the same equipment. If all N-nodes and S-nodes are merged, it becomes a non-sharing network.

The objective is to compute the subscription plans of the TCC links given certain pricing models and the N-node locations so as to transmit the data of all S-nodes, with a minimization of the overall monetary cost. SAMS works on vendor-responsible commercial equipment. Therefore, the vendor has the knowledge of all its responsible equipment in a region and can optimize in a centralized fashion. As the very first work, we assume in this paper that 1) our context is one vendor only, 2) N-nodes will not relay traffic for other N-nodes, 3) S-nodes will not relay traffic for other S-nodes, and 4) there is no in-network processing of the traffic. We plan to study routing of other forms, intermediate traffic caching, coding, etc., and multi-vendor joint optimization in our future work.

D. Related Work

With the emerging IoT applications such as SAMS, we are the first to propose the ICS model. There are studies on network multiplexing/sharing in other *context*. On the ISP-side, there are resource managements for 3G traffic flows, in particular for multimedia data traffic. There are caching [14], prefetching [15], framing [16], etc., to serve requests in some forms of aggregation. The computation is in the e-nodeB managed by the ISPs. In the ICS model, the optimization is managed by the vendor facing certain ISP pricing models. On the client side, the hotspot function is occasionally used to share 3G data, primarily for convenience. Family plans also exist. These client side sharing are spontaneous, and there is barely any intentional coordination, which is needed in our context.

We further comment on two foundational networking paradigm Wireless Sensor Networks (WSN) [17] and Fog Computing [18], [19]. In WSN, since wireless sensors are energy constrained and communication dominates energy consumption, the optimization objective is on all communication links within the WSN. The constraint in the SAMS network model is the TCC links between the things and clouds. Thus, we differ from WSN in the *optimization objective*. In addition, the pricing model of the TCC links also differs from the cost model of the WSN links. The idea of Fog Computing is to relocate functions to the edge, either for a fast response or for cost saving. Fog Computing is a conceptual framework. SAMS represents *concrete* application scenarios and can be regarded as one instance of Fog Computing.

III. Problem Formulation and Complexity Analysis

In this section, we first present the network settings. Then we formulate the ICS problem as a cost minimization problem. Finally, we analyze the problem complexity.

A. Network Topology

The considered network includes m S-nodes and q N-nodes. Let $\mathcal{N} = \{n_1, n_2, ..., n_q\}$ denote the set of N-nodes. An N-node can be either *installed* or *vacant*. Let f_j denote the indicator, i.e., $f_j = 1$ if installed; $f_j = 0$ if vacant. We define $\mathbf{f} \triangleq (f_1, f_2, \ldots, f_q)$. $\mathbf{f}$ is a decision variable to be optimized.

Let $\mathcal{S} = \{s_1, s_2, ..., s_m\}$ be the set of S-nodes. Let $\mathcal{S}_j$ denote the subset of S-nodes, which can reach N-node n_j. We define $\mathbb{S} \triangleq \{\mathcal{S}_1, \mathcal{S}_2, \ldots \mathcal{S}_q\}$. Here, the term "reach" means that it is possible for the S-node to deliver its data to the N-node through some LOC links. We assume that $\bigcup_{j=1}^{q} \mathcal{S}_j = \mathcal{S}$, i.e., each S-node can reach at least one N-node. One of our design aims is to install a subset of N-nodes to cover all S-nodes, i.e.,

$$\bigcup_{j: f_j = 1} \mathcal{S}_j = \mathcal{S}. \tag{1}$$

In the network, the thing-to-cloud communication (TCC) links connect N-nodes and the cloud, which are charged by ISPs. LOC links connect S-node and N-nodes, which are free.

B. Load Constraint Modeling

In this subsection, we discuss how each N-node is able to accommodate the data usage from its connected S-nodes. In each billing cycle (e.g., one month), S-node s_i requires to upload a data volume of u_i to the cloud. The S-node's data volume is split to be transferred via one or more N-nodes. We define $\mathbf{u} \triangleq (u_1, u_2, \ldots, u_m)$. Let v_{ij} be the split data volume of s_i transferred via n_j. We define $\mathbf{v} \triangleq (v_{ij} : \forall i = 1, 2, \ldots m, \forall j = 1, 2, \ldots, q)$. In this paper, $\mathbf{u}$ is given as a priori, and $\mathbf{v}$ is to be optimized. We assume that u_i and v_{ij} values are integers (e.g., in kilobytes).

For the given u_i for S-node s_i, v_{ij} are decision variables to be designed. Since s_i's data are transferred via its connected installed N-nodes, we have

$$\sum_{j: f_j = 1} v_{ij} = u_i, \forall i = 1, 2, \ldots, m. \tag{2}$$

[3] In some regions, ISPs are developing WiFi networks. Note that such network cannot be free either. ISPs will not develop free infrastructure to serve hundreds of commercial vendors in large scale. As such, such networks are to be subscribed and can be considered as TCC links as well.

The data of s_i cannot be uploaded via N-nodes that are vacant or out of its communication range. Therefore, we have

$$v_{ij} = 0, \forall i, j : f_j = 0 \vee s_i \notin \mathcal{S}_j. \tag{3}$$

In each billing cycle, by paying to the ISP, each N-node could purchase a data volume allowed to upload to the cloud via its TCC link. Let d_j denote the data volume purchased by n_j. We define $\mathbf{d} \triangleq (d_1, d_2, \ldots, d_q)$. $\mathbf{d}$ is a decision variable. If n_j is installed, the load w_j at the TCC link of n_j is the accumulated data amount uploaded by its connected S-nodes. Therefore, we have $w_j = \sum_{i:s_i \in \mathcal{S}_j} v_{ij}$. $w_j = 0$ if n_j is vacant. The load w_j at the TCC link of node n_j cannot exceed the purchased data volume d_j. Therefore, we have

$$\sum_{i:s_i \in \mathcal{S}_j} v_{ij} \leq d_j, \forall j = 1, 2, \ldots, q. \tag{4}$$

C. The Cost of TCC Sharing

For TCC links, let $\mathcal{C} = \{c_1, c_2, \ldots, c_l\}$ be the set of available plans provided by ISPs, where k denotes the number of plans. The monetary cost for data volume x of plan c_i $(i = 1, 2, \ldots, l)$ can be presented as a function $c_i(x)$.[4]

Let $c'_j \in \mathcal{C}$ denote the plan adopted by N-node n_j. We define $\mathbf{c}' \triangleq (c'_1, c'_2, \ldots, c'_q)$. Once a plan is selected, it cannot be changed within the billing cycle. For the N-node n_j with purchased data volume d_j, the communication cost of n_j can be present as $c'_j(d_j)$. Thus, the total communication cost for q N-node locations is given as:

$$T_{total} = \sum_{j=1}^{q} c'_j\left(d_j\right). \tag{5}$$

D. IoT Communication Sharing Problem Formulation

The goal of the IoT Communication Sharing (ICS) problem in this paper is to minimize the overall monetary cost of the TCC links, given a set of available plans, the network topology, and the possible locations for S-nodes and N-nodes. Hence, the overall monetary cost of the TCC links is the objective function. The decision variables are the installation indicators $\mathbf{f}$, the adopted cost function $\mathbf{c}'$, the subscribed data volume $\mathbf{d}$ and the data volume $\mathbf{v}$ from each S-node s_i uploaded from each N-node n_j. The constraints are shown in Sections III-A and III-B. In summary, we have the following optimization problem:

Problem 1. *(ICS) Given $\mathcal{S}$, $\mathcal{N}$, $\mathbf{u}$ and $\mathcal{C}$, determine $\mathbf{f}$, $\mathbf{c}'$, $\mathbf{d}$ and $\mathbf{v}$, subject to constraints (1), (2), (3) and (4), to minimize $T_{\text{total}} = \sum_{j=1}^{q} c'_j\left(d_j\right)$.*

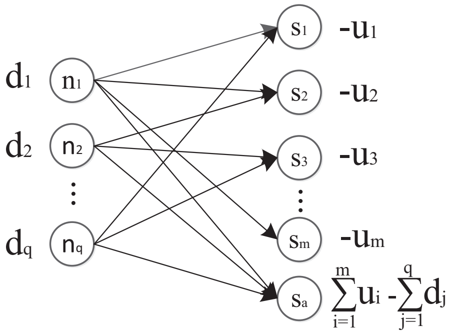

Fig. 4: Illustration of conversion to minimum cost flow problem.

E. Problem analysis

Theorem 1. *Problem ICS is NP-complete.*

Proof. We prove this theorem by transforming the problem into the minimum set cover problem. Consider a special case that $u_i = 1, \forall i = 1, 2, \ldots, m$. Let the monetary cost at each N-node be a if it is installed, be 0 if vacant. Therefore, we aim to minimize the number of installed N-nodes. As a result, Problem 1 is equivalent to an optimal set cover problem: to select a minimum number of sets from $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_q\}$ that covers all elements in the $\mathcal{S}$. □

IV. The ICS Algorithm

In this section, we solve the ICS problem. However, ICS problem is NP-complete, it is unrealistic to find a globally optimal solution within polynomial time. We design Minimize Communication Cost algorithm (denoted as MCC) which achieves a locally optimal solution. The rationale to develop MCC is as follows. We need to determine 1) the purchasing strategy, i.e., $\mathbf{c}'$ and $\mathbf{d}$ values, and 2) the data upload scheme $\mathbf{v}$. Accordingly, we develop two sub-functions: `best-Plan()` and `best-Upload()`. Given the data upload scheme $\mathbf{v}$, `best-Plan()` will search for the minimized cost purchasing strategy i.e., $\mathbf{c}$ and $\mathbf{d}$. Given $\mathbf{c}$ and $\mathbf{d}$, `best-Upload()` will find an even better data upload scheme $\mathbf{v}$. `best-Plan()` and `best-Upload()` are then conducted alternatively to gradually improve the overall cost.

Given the data upload scheme $\mathbf{v}$, finding the best purchasing strategy (i.e, `best-Plan()`) can be optimally solved. `best-Plan()` first computes the data usage of each N-node according to $\mathbf{v}$. Then, knowing the data usage, it can find the minimized cost purchased strategy for each N-node by simply searching within all plans in $\mathcal{C}$.

Given the purchasing strategy (i.e., $\mathbf{c}'$ and $\mathbf{d}$), finding the best data upload scheme $\mathbf{v}$ (i.e., `best-Upload()`) can be converted to the minimum cost flow problem. We illustrate the conversion in Fig. 4. We construct a graph which contains q N-nodes, m S-nodes and one auxiliary S-node as the vertices. If $s_i \in \mathcal{S}_j$, then an edge is added between s_i and n_j. For the auxiliary S-node, we add edges between it and all N-nodes. For N-node n_j, we attach the (positive) purchased data volume d_j to it. For each S-node s_i, we attach the (negative) data usage $-u_i$ to it. For the auxiliary S-node s_a, we attach the negative

[4] For different pricing models, the form of cost functions are different. For example, the monthly plan (MP) pricing model, ISPs provide a set of monthly data plans. For data plan m_i, let t_i denote the price charged for the fixed amount of cap usage(denoted as k_i). If the data usage hits this cap then a higher price c is charged for each per data usage unit above the cap, the cost function can be presented as

$$c_{m_i}(x) = \begin{cases} t_i & , x \leq k_i \\ t_i + c(x - k_i) & , x > k_i. \end{cases}$$

The pay-as-you-go pricing model is discussed in Section V.

data usage $\sum_{i=1}^{m} u_i - \sum_{j=1}^{q} d_j$ to it. For the edges connected to n_j, the attached unit delivery costs are $\frac{c'_j(d_j)}{d_j}$. Our problem now is equivalent to obtaining the minimum cost flow through the network. The N-nodes are "sources" for the flow entering the system and the S-nodes are "sinks" where flow leaves the system. Brenner's algorithm [20] is used in `best-Upload()` to solve the minimum cost flow problem with the computational complexity of $\mathcal{O}(q(\log q)^2(m+1)^2)$.

Algorithm 1 Minimize Communication Cost, `MCC`$(\mathcal{S}, \mathcal{N}, \mathbb{S}, \mathbf{u}, \mathcal{C})$.

```
1: Initialize $\mathbf{v} \leftarrow \mathbf{0}, \mathbf{f} \leftarrow \mathbf{0}, \mathbf{c}' \leftarrow \mathbf{0}, \mathbf{d} \leftarrow \mathbf{0}, \widetilde{\mathbf{v}} \leftarrow \mathbf{0}$
2: $\mathbf{v} \leftarrow$ init-Upload($\mathcal{S}, \mathcal{N}, \mathbb{S}, \mathbf{u}$)
3: repeat
4:     $\widetilde{\mathbf{v}} \leftarrow \mathbf{v}$
5:     $[\mathbf{c}', \mathbf{d}] \leftarrow$ best-Plan($\mathcal{C}, \mathbf{v}$)
6:     $\mathbf{v} \leftarrow$ best-Upload($\mathcal{S}, \mathcal{N}, \mathbb{S}, \mathbf{u}, \mathbf{c}', \mathbf{d}$)
7: until $\widetilde{\mathbf{v}} == \mathbf{v}$
8: $\mathbf{f} \leftarrow$ compute-Indicator($\mathbf{d}$)
9: return $\mathbf{c}, \mathbf{d}, \mathbf{f}, \mathbf{v}$
```

The overall algorithm `MCC()` is an iterative algorithm shown in Algorithm 1. The overall algorithm first calls `init-Upload()` (line 2) to initialize each $v_{ij} \in \mathbf{v}$ to u_i and then it calls `best-Plan()` (line 5) to determine the purchasing strategy $\mathbf{c}$ and $\mathbf{d}$. Such $\mathbf{c}$ and $\mathbf{d}$ are given to `best-Upload()` (line 6). `best-Upload()` will adjust the data upload scheme $\mathbf{v}$ according to the purchasing strategy. Such $\mathbf{v}$ is returned to `best-Plan()`. The termination condition for iteration is that there is no change in the data upload scheme $\mathbf{v}$. Then `MCC()` calls `compute-Indicator()` to compute the installed/vacant indicator of N-nodes (line 8) according to $\mathbf{d}$ (if $d_j > 0$, then $f_j = 1$. Otherwise $f_j = 0$).

We now analyze the convergence of `MCC()`. Let $\mathbf{d}_y$ denote the purchased data volume $\mathbf{d}$ result of the `best-Plan()` in the y-th round iteration. The maximum value of each entry of $\mathbf{d}$ is limited to $d_{\max}$. According to Brenner's algorithm, $\mathbf{d}_{y+1} \preceq \mathbf{d}_y$ when `best-Plan()` is called. Since $\mathbf{d}$ is bounded (i.e., $\mathbf{0} \preceq \mathbf{d}$), $\mathbf{d}_y$ will converge after a finite number of iterations. In each iteration at least one entry of $\mathbf{d}$ is decreased by 1. Therefore, the algorithm will converge at most in r rounds, to a local optimal, where $r = qd_{\max}$. `MCC()` is a polynomial-time algorithm with the computational complexity of $\mathcal{O}(rq(\log q)^2(m+1)^2 + rql)$.

V. ICS in the Pay-As-You-Go Pricing Model

We now specifically consider the pay-as-you-go (PAYG) pricing model. This is because PAYG is likely to be the primary pricing model for IoT communication for this moment when the IoT industry is still in its early stage. Looking into the history, PAYG is always the pricing model in early stages of a new business, e.g., pay per call, pay per megabyte of data. Monthly plan (MP) emerges when the business becomes mature and as a mean of price reduction when facing competition [21]. As a matter of fact, in our experiment, the CAT1's pricing model is PAYG.

A. Problems

For the PAYG pricing model, $\mathcal{C} = \{c_1\}$, i.e., there is only one plan for PAYG. The cost function is represented by Eq. (6). This is a staircase function. Here x is the data usage; L is an integer to denote the step size of pricing model. Let p_i be the price for the i-th step of L data volume. In practice, p_i decreases as the price step increases [22] and $\lim_{i \to +\infty} p_i = p_{\min}$, where $p_{\min}$ is positive.

$$c_1(x) = \sum_{i=1}^{\lceil \frac{x}{L} \rceil} p_i. \tag{6}$$

The overall cost using PAYG is $\sum_{j=1}^{q} c_1 \left(\sum_{i:s_i \in \mathcal{S}_j} v_{ij} \right)$. Thus, we arrive the following problem:

Problem 2 (ICS-PAYG)**.** *Given $\mathcal{S}$, $\mathcal{N}$, $\mathbb{S}$, $\mathbf{u}$ and c_1, determine $\mathbf{f}$, $\mathbf{d}$ and $\mathbf{v}$, subject to constraints (1), (2), (3) and (4), to minimize $T_{\text{total}} = \sum_{j=1}^{q} c_1 \left(\sum_{i:s_i \in \mathcal{S}_j} v_{ij} \right)$.*

In reality, we notice that many S-nodes can reach a limited number of N-nodes. We therefore consider a case that the degree of an S-node (i.e., the number of N-nodes an S-nodes can reach) is limited to D. We have the following problem:

Problem 3 (ICS-D-PAYG)**.** *Given $\mathcal{S}$, $\mathcal{N}$, $\mathbb{S}$, $\mathbf{u}$, D and c_1, determine $\mathbf{f}$, $\mathbf{d}$ and $\mathbf{v}$, subject to constraints (1), (2), (3) and (4), to minimize $T_{\text{total}} = \sum_{j=1}^{q} c_1 \left(\sum_{i:s_i \in \mathcal{S}_j} v_{ij} \right)$.*

Theorem 2. *Problems ICS-PAYG and ICS-D-PAYG are both NP-complete.*

The proof is similar to the proof of Theorem 1. Intrinsically, the complexity comes from N-nodes covering S-nodes, rather than the pricing model; thus, the complexity of NP-completeness holds.

B. Algorithms

We develop Fast N-node Deployment (FND) algorithm for the ICS-PAYG problem, and Layering N-node Deployment (LND) algorithm for the ICS-D-PAYG problem.

The problem ICS-PAYG and ICS-D-PAYG can be divided into three subproblems: N-node placement to cover all S-nodes (i.e. to find $\mathbf{f}$), the upload scheme (i.e. to find $\mathbf{v}$) and data volume subscription at each installed N-node (i.e. to find $\mathbf{d}$).

FND and LND first solve the N-node placement to cover all S-nodes and the upload scheme. The TCC link placement to cover all S-nodes is a set cover problem. For the ICS-PAYG problem, FND adopts the greedy set cover algorithm in [23]. For the ICS-D-PAYG problem, LND employs the layering set cover algorithm in [21] which takes advantage of the degree information of S-nodes. In this way, the greedy set cover algorithm and the layering set cover algorithm select N-node one by one. Whenever an N-node is selected, the newly covered S-nodes will upload all their data volume via this N-node.

After the above steps to determine $\mathbf{f}$ and $\mathbf{v}$ for both FND and LND, each installed N-node subscribes the closest data cap that is greater than the data volume needed to be transferred via it, so that $\mathbf{d}$ is determined.

Theorem 3. *The approximation ratio of the algorithm FND for ICS-PAYG is* $\frac{p_1}{p_{\min}}(\ln m + 2)$.

Proof. Let the cost of FND be r_{f}, and the optimal cost of ICS-PAYG (denoted as OPT) be r_{o}. Directly proving $\frac{r_{\mathrm{f}}}{r_{\mathrm{o}}} \leq \frac{p_1}{p_{\min}}(\ln m + 2)$ is hard, we prove this by divide and conquer. As the cost is related to the number of installed N-node and the number of purchased data volume steps under the PAYG pricing model, we first prove the installed N-node number of FND (denoted as k_{f}) and OPT (denoted as k_{o}) meets $\frac{k_{\mathrm{f}}}{k_{\mathrm{o}}} \leq \ln m + 1$, then we prove the purchased steps of FND (denoted as t_{f}) and OPT (denoted as t_{o}) meets $t_{\mathrm{f}} \leq t_{\mathrm{o}} + k_{\mathrm{f}}$. Based on these, we can prove $\frac{r_{\mathrm{f}}}{r_{\mathrm{o}}} \leq \frac{p_1}{p_{\min}}(\ln m + 2)$.

We first prove $\frac{k_{\mathrm{f}}}{k_{\mathrm{o}}} \leq \ln m + 1$. Let $k_{\min}$ be minimum number of N-nodes that can cover all S-nodes. We have $\frac{k_{\min}}{k_{\mathrm{o}}} \leq 1$ and $\frac{k_{\mathrm{f}}}{k_{\min}} \leq \ln m + 1$ (by the approximation ratio of greedy set cover algorithm). Then we have $\frac{k_{\mathrm{f}}}{k_{\mathrm{o}}} \leq \ln m + 1$.

We then prove $t_{\mathrm{f}} \leq t_{\mathrm{o}} + k_{\mathrm{f}}$. Let X denote the total data usage of all S-nodes, and $x = Lq - r$, where $q \in \mathbb{N}, 0 \leq r < L$. $q \leq t_f$, $q \leq t_o$ (otherwise the purchased data volume of FND and OPT is smaller than total data usage x). Let x_j ($j = 1, 2, \ldots, k_{\mathrm{f}}$) denote the data usage of the installed N-node j of FND, and $x_j = Le_j - r_j$, where $e_j \in \mathbb{N}, 0 \leq r_j < L$. e_j is the steps purchased by the installed N-node j, thus $t_{\mathrm{f}} = \sum_{j=1}^{k_{\mathrm{f}}} e_j$, $x = qL - r = L\sum_{j=1}^{k_{\mathrm{f}}} e_j - \sum_{j=1}^{k_{\mathrm{f}}} r_j$, we have $t_{\mathrm{f}} = q - \frac{r}{L} + \frac{\sum_{j=1}^{k_{\mathrm{f}}} r_j}{L}$, as $0 \leq r_j < L$, we have $t_{\mathrm{f}} \leq q + \frac{k_{\mathrm{f}} L}{L} = q + k_{\mathrm{f}}$.

At last, we prove $\frac{r_{\mathrm{f}}}{r_{\mathrm{o}}} \leq \frac{P_1}{P_{\min}}(\ln m + 2)$ based on the above proof. The price of each step ranges from p_1 to $p_{\min}$, thus $r_{\mathrm{f}} \leq p_1 t_{\mathrm{f}}$ and $r_{\mathrm{o}} \geq p_{min} t_{\mathrm{o}}$. $\frac{r_{\mathrm{f}}}{r_{\mathrm{o}}} \leq \frac{p_1 t_{\mathrm{f}}}{p_{\min} t_{\mathrm{o}}} \leq \frac{p_1}{p_{\min}}(\frac{t_o}{t_o} + \frac{k_f}{t_o})$, each installed N-node must purchase at least one step, thus $t_o \leq k_o$, so $\frac{r_{\mathrm{f}}}{r_{\mathrm{o}}} \leq \frac{p_1}{p_{\min}}(1 + \frac{k_{\mathrm{f}}}{k_{\mathrm{o}}}) \leq \frac{p_1}{p_{\min}}(\ln m + 2)$. □

Theorem 4. *The approximation ratio of the algorithm LND for ICS-D-PAYG is* $\frac{p_1}{p_{min}}(D + 1)$.

The difference between ICS-PAYG and ICS-D-PAYG comes from TCC link placement to cover all S-nodes which is equivalent to the minimum set cover problem. Different with the ICS-PAYG problem, the maximum degree of S-nodes D is given in the ICS-D-PAYG problem, thus the information about the degree of S-nodes can be utilized to improve N-node placement. Using the degree information, the number of installed N-nodes selected by LND is at most D times to the minimum set cover, while this ratio is $\ln m + 1$ in FND for ICS-PAYG problem.

FND and LND are based on greedy algorithms. Though they have bounded performance, they may not perform well in some special cases. If that happens, we also develop a heuristic `MCC-PAYG()` based on `MCC()`. As the pricing model is determined, there is no need to search plans for N-nodes. Thus, line 5 of `MCC()` can be replaced by computing the subscription data volume directly. The complexity of `MCC-PAYG()` is $\mathcal{O}(rq(\log q)^2(m+1)^2)$.

VI. Implementation

A. Hardware and Communication Link Choices

For S-node and N-node, we use Raspberry Pi 3 Model B as the hardware board (Fig. 5 and Fig. 6). We connect S-node and a Fan. A Fan has a pulse width modulation (PWM) interfaces that can be used to connect the digital I/O ports of the Raspberry Pi using DuPont cables. The Fan data are periodically sent to PWM, and then to the S-node. Other equipment such as chillers and pumps have standard APIs to output data from their embedded sensors. Using chiller as an example, a chiller controller uses a ModBus RTU protocol with an RS-485 interface. Modbus RTU protocol is a query-response protocol. We implement an application in Raspberry Pi using the standard library libmodbus [24] to query the chiller through Modbus RTU protocol. The communication between USB port of Raspberry Pi and RS485 need a USB/RS485 Converter module as the electrical level difference.

For the LOC of both the S-node and N-node, we use a Texas Instruments CC 2560 SimpleLink$^{\mathrm{TM}}$ Wireless MCU for the 802.15.4 radio interface. Then this module is connected to Raspberry Pi using a USB-to-serial cable.

For the TCC side of the N-node, as the interface of Raspberry Pi is TTL, while the interface provided by CAT1 is RS-232, we use the MAX3232 as a converter. We rent CAT1 data plans from Telecom Anonymity.

We rent a server in Cloud Anonymity with 8 cores of 2.5 GHz, and a total memory of 128 GB.

B. The Network Format Choice

We choose 6LoWPan (IPv6) as the network layer protocol. We overcame two implementation challenges.

The first is that the ISP provided CAT1 only supports IPv4. Moreover, it only provides application layer interfaces. Thus, we develop an IPv6-IPv4 converter. It locates in the application layer of the N-node, yet it emulates the network layer. It has two functions: packet format transformation and IPv6-IPv4 address mapping. For the IPv6 packet we get from the LOC network, we remove all headers to get the application packet. Then we put such packet to the CAT1 interface. The address mapping is done by mapping a group of IPv6 address to an IPv4 address (the address of CAT1) and a port number. Every N-node establishes a translation table of the mapping. Each entry in this translation table is automatically inserted when the first packet from the S-node reaches the N-node, i.e., N-node allocates each S-node connected to it a universal port number with the CAT1's IPv4 address.

The second challenge is that in practice, an S-node should have a fixed IP address. We do not have fixed IP addresses. In our implementation, each S-node gets its IPv6 address from N-node using the uIP library from Contiki, making the IP address dynamic. Since the interaction between an S-node and the cloud is bi-directional, the dynamic IP address can break the interaction. To this end, in the application layer, we develop a notification mechanism such that if the IP address of the S-node changes, the S-node will notify the cloud.

C. The Routing Choice

In our IoT application context, data are routed from the S-nodes to the cloud. We choose RPL [25] for routing. RPL is a gradient routing technique that organizes nodes as a Direct Acyclic Graph (DAG) rooted at the sink. RPL has an objective

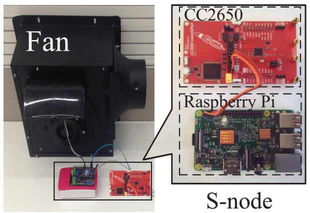

Fig. 5: Fan with Raspberry Pi S-node.

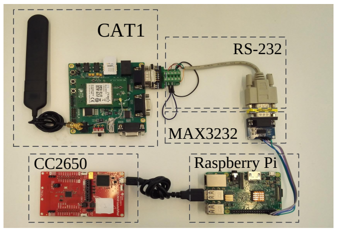

Fig. 6: The N-node.

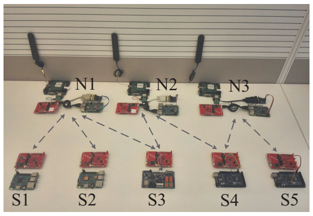

Fig. 7: The network topology of the experiments.

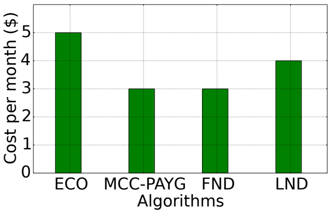

Fig. 8: The monthly cost of different algorithms.

function. The goal is to minimize the cost to reach the sink from any node. This function has to be customized. Recall that in our algorithm, we compute the amount of traffic an S-node sends to each peering N-node. In our implementation, the objective function maintains a "volume-N-node" table. The table records the residual data volume of the S-node can be transmitted through its peering N-node. The objective function chooses the N-node with residual data volume in a round-robin fashion. For each S-node, the initial data volume of its peering N-nodes is computed by `MCC()`(in Section IV).

VII. Performance Evaluation

A. Evaluation by Experiments

1) System setup: The network topology is shown in Fig. 7. There are three N-nodes and five S-nodes. The links are configured as in the figure. We set the data traffic for S_1 and S_2 to be 200 bytes every three minutes, and the data traffic for S_3, S_4 and S_5 to be 600 bytes every minute.[5] We adopt the PAYG model from TeleCom Anonymity. Each 40 MB costs $1, i.e., $L = 40$, $p_1 = p_2 \ldots = p_{\min} = 1$ in Eq. (6).

We compare three IoT communication sharing algorithms MCC-PAYG, FND and LND with the exclusive channel occupation (ECO) algorithm, i.e., each device transmits its data directly to the cloud via its dedicated purchased TCC link.

2) Experiment Results: The system is turned on for 6 hours and the overall data usage is scaled to one month. We derive the overall monthly cost of different algorithms. The results are shown in Fig. 8. We can observe that under the PAYG model, MCC-PAYG, FND and LND lead to a cost saving of 20%–40% as compared with ECO. This matches our expectation since IoT communication sharing will bring significant cost reductions. Next, we will evaluate a variety set of network configurations by trace-driven simulations, and we will see that the saving is more significant when the network is larger.

B. Evaluation by Trace-driven Simulations

We now use trace-driven simulations to evaluate ICS. We first present two real-world cases and price models employed in the evaluation. We then show the result of the cost reduction introduced by ICS. We also show how ICS effectively makes use of the purchased data volume of TCC link, and the performance of algorithms for the PAYG pricing model.

1) Simulation Setup: We evaluate ICS and algorithms by simulation using two real-world cases.

Case 1: We work on a SAMS application and we collect data of building group belong to Anonymity property Ltd (denoted as B1). It consists of three buildings (denoted as B1-one, B1-two and B1-three). We collect the data of chillers and pumps which belong to a same vendor Anonymity. The chillers and the pumps of each building are located in the plant room on the top floor. The number of the chillers and the pumps of each building of B1 is showed in Table I. Each equipment connects to an S-node and can be regarded as a possible location of N-node. The network topology of B1 is shown in Fig. 9.

We collected four types of data for the chiller to compute COP (Section II), such as the supplying/returning chilled water temperature. We collect data of the power input and the heat transfer to circulating water for the pump to compute Water Transfer Coefficient (WTC) [6]. The data of the chillers and the pumps were collected at 30 minute intervals. The monthly data volume collected from each chiller and each pump are showed in Table II.

Case 2: We also employ the publicly available data. We use the data of a building (denoted as B2) located at Kuwait University [26]. The data are from chillers, pumps, air handling units (AHU) and cooling towers of B2. The number of each kind of equipment is showed in Table I. Each equipment can be regarded as an S-node and a possible location of N-node. The network topology of B1 is shown in Fig. 10.

For the traffic of chiller, pump and cooling tower, we use the real traffic collected in [27]. The data of chiller, pump and cooling tower were collected at one minute intervals. The data of chillers includes 8 types of data, such as work load, supply temperature and return temperature. For the traffic of AHU, we use the real traffic pattern described in [28], which also collects data at one minute intervals. The data of AHU includes 14 features including air mass flow rate, room air temperature, etc. The monthly data volume of the four types of equipment is shown in Table II.

The pricing models: We study two pricing models: 1) PAYG, the first 40 MB costs $1, i.e., $L = 40, p_1 = 1$, the prices of the following 40 MB steps are $0.8, i.e., $p_2 = p_3, \ldots = p_{\min} = 0.8$ in Eq. (6); 2) MP, the monthly data plans are shown in Table III with $0.6 charged for each 1 MB over the cap.

Evaluation Criteria: We evaluate ICS employing MCC algorithm and ECO with the settings of in Case1 and Case2

[5] Our S-nodes and N-nodes do not connect to equipment, since 1) our IoT communication sharing is general for all types of equipment, and 2) we admit that we do not have enough Fans/Chillers for an eight-node experiment.

[6] WTC is a performance index of the pump. The pump should be maintained before the WTC under a threshold.

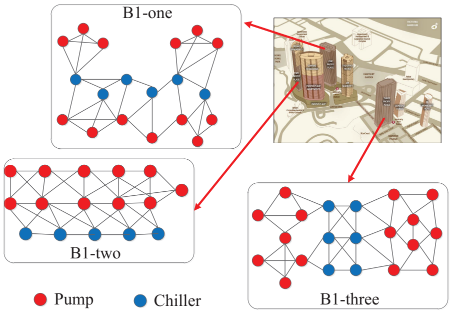

Fig. 9: Illustration of the topology of B1.

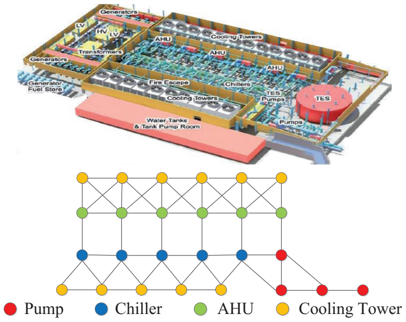

Fig. 10: Illustration of the topology of B2.

Equipment number	B1			B2
	one	two	three	
Chiller	6	5	6	5
Pump	13	11	15	4
AHU	-	-	-	6
Tower	-	-	-	11

TABLE I: The number of equipment at B1 and B2.

TABLE II: Monthly data usage of different equipment.

	Chiller	Pump	AHU	Tower
B1	6.62 MB	4.25 MB	-	-
B2	9.26 MB	3.65 MB	12.45 MB	3.90 MB

under the two pricing models. We also run MCC-PAYG, FND and LND algorithms for PAYG pricing model.

We evaluate the data transmission cost of ICS and ECO. We introduce *underutilized ratio* for TCC link. The underutilized ratio of TCC link is the ratio between the unused purchased data volume and the purchased data volume of TCC link. The underutilized ratio can indicate how effectively the purchased data volume of TCC link has been used. We evaluate the underutilized ratio of TCC link of ICS and ECO. We also evaluate the data transmission cost of MCC-PAYG, FND and LND under PAYG pricing model.

2) Results: **Data Transmission Cost Reduction:** We first compare ECO and ICS under two pricing models at B1 in Figs. 11–12. We see that ICS shows a higher cost saving compared with experimental results. This matches our expectation since the advantage of sharing becomes more significant when there are more S-nodes to share. For the PAYG pricing model, the cost of ECO is 4.8 times, 8 times and 4.4 times to that of ICS at B1-one, B1-two and B1-three respectively. For the MP pricing model, the cost of ECO is 3.2 times, 3.2 times and 2.4 times to that of ICS at B1-one, B1-two and B1-three respectively, a slightly less than that of PAYG. This is because, in MP pricing model, the cost gap between two adjacent plans is bigger, thus if the data volume of one monthly data plan cannot meet the requirement of an N-node, the N-node has to purchase the other one whose price is much higher so that purchased data are underutilized, while in the PAYG model, the TCC link can purchase steps which are cheaper one by one.

In Fig. 13, we also compare the cost of ECO and ICS under the PAYG and MP pricing models at B2. We see cost savings of 78% and 71% on PAYG and MP respectively. This further confirms that ICS significantly outperform ECO.

The underutilized ratio of TCC link: We compare the underutilized ratio of TCC links under ECO and ICS. Please note that the higher underutilized ratio indicates the customer waste more data volume which has been paid. High underutilized data volume ratio discourages customers.

In Figs. 14–15, we show the cumulative distribution function (CDF) of TCC links' underutilized ratio of ECO and ICS at B1 and B2 under the PAYG pricing model where $L = 10$ MB. In Fig. 14, we can observe that 20% TCC link's underutilized ratio of ICS at B1 is 0% which means the data volume of these TCC links has been used up without waste. No TCC links' purchase can be fully used under ECO. We can also observe that the underutilized ratio of all TCC links of ICS is under 23%. For the ECO, the underutilized ratios of TCC links could be as much as 57%.

The underutilized ratio gap between ECO and ICS is even greater at B2, shown in Fig. 15. We can see that 50% TCC links' underutilized ratio is 0%, while this value is still 0 for ECO. We can also observe that the underutilized ratio of all TCC links of ICS is under 18%. For the ECO, the underutilized ratio of TCC links could be as much as 61%. This illustrates that through IoT communication sharing, the purchased data volume of TCC link can be made better use of compared with ECO. This is also the reason why ICS can lead to a substantial cost reduction compared with ECO.

The performance of algorithms for PAYG: We compare our MCC-PAYG algorithm with FND and LND algorithms for the PAYG pricing model.

In Figs. 16–17, we show the monthly cost of MCC-PAYG, FND and LND at B1 and B2 respectively. We notice that MCC-PAYG, FND and LND have a cost saving of 76%–88% compared with ECO (the costs of ECO at three building of B1 shown in Fig. 11 are \$19, \$16 and \$21 respectively, and \$26 at B2 shown in Fig. 13). These results illustrate that MCC-PAYG, FND and LND work effectively compared with ECO under the PAYG pricing model.

We see that MCC-PAYG outperforms FND and LND in these two situations. We also see that MCC-PAYG and FND outperform LND. Compared with LND, FND has cost saving from 26% to 50%. This is because, at B1 and B2, the maximum degree of S-nodes range from 5 to 8, and LND is suitable for the scenario where the degree of S-nodes is small.

VIII. Conclusion

In this paper, we carefully analyzed the emerging IoT applications such as smart after-sales maintenance and services. We showed that a separate IoT network is needed to serve their requirement of sending the data to the cloud. A core obstacle is the high costs of communication choices. We proposed IoT communication sharing (ICS), which can effectively reduce cost in an order. We then presented a comprehensive study of the ICS

Type	Price	Volume
1	2 $	30 MB
2	5 $	100 MB
3	7 $	200 MB
4	15 $	500 MB
5	25 $	1 GB

TABLE III: The monthly data plans.

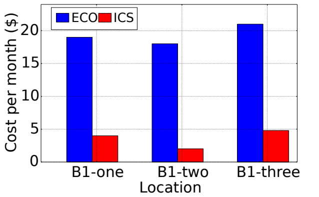

Fig. 11: The monthly cost at B1 under the PAYG pricing model.

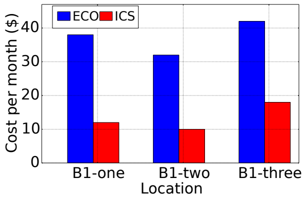

Fig. 12: The monthly cost at B1 under the MP pricing model.

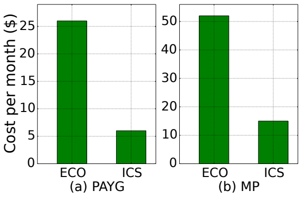

Fig. 13: The monthly cost at B2 under different pricing model.

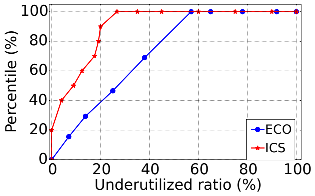

Fig. 14: The CDF of the underutilized ratio of TCC links at B1.

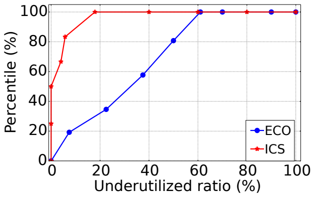

Fig. 15: The CDF of the underutilized ratio of TCC links at B2.

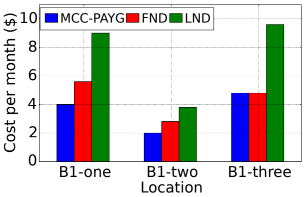

Fig. 16: The monthly cost of different algorithms for PAYG at B1.

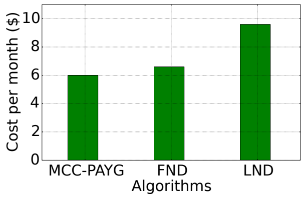

Fig. 17: The monthly cost of different algorithms for PAYG at B2.

problem, with problem formulation, algorithms, implementation and evaluation using experiments and trace-driven simulations for two real world cases.

IX. ACKNOWLEDGMENTS

We would like to acknowledge the support of the National Natural Science Foundation of China under Grant 61272464 and RGC/GRF under Grant PolyU 5264/13E. This work was also supported by the University of Sydney DVC Research/Bridging Support Grant.

REFERENCES

[1] D. Niyato, X. Lu, P. Wang, D. I. Kim, and Z. Han, "Economics of Internet of Things: an information market approach," *IEEE Wireless Communications*, vol. 23, no. 4, pp. 136–145, 2016.

[2] A. M. Vegni, V. Loscr, A. Neri, and M. Leo, "A Bayesian packet sharing approach for noisy IoT scenarios," in *Proc. IEEE IoTDI'16*, Berlin, Germany, Apr. 2016.

[3] Z. Zheng, D. Wang, J. Pei, Y. Yuan, C. Fan, and F. Xiao, "Urban traffic prediction through the second use of inexpensive big data from buildings," in *Proc. ACM CIKM'16*, Indianapolis, IN, Oct. 2016.

[4] J. Lee, C. Jin, and Z. Liu, "Predictive big data analytics and cyber physical systems for TES systems," in *Advances in Through-life Engineering Services*. Springer, 2017, pp. 97–112.

[5] C. A. Garc, P. Merino *et al.*, "3GPP standards to deliver LTE connectivity for IoT," in *Proc. IEEE IoTDI'16*, Berlin, Germany, Apr. 2016.

[6] G. Naddafzadeh-Shirazi, L. Lampe, and G. Vos, "Coverage enhancement techniques for machine-to-machine communications over LTE," *IEEE Communications Magazine*, vol. 53, no. 7, pp. 192–200, 2015.

[7] J. Roessler, "LTE-advanced (3GPP rel. 12) technology introduction," Apr. 2015, White Paper. [Online]. Available: https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1ma252/1MA252_2e_LTE_Rel12_technology.pdf

[8] J. H. Lai, F. W. Yik, and A. K. Chan, "Maintenance cost of chiller plants in hong kong," *Building Services Engineering Research and Technology*, vol. 30, no. 1, pp. 65–78, 2009.

[9] N. Firdaus and B. T. Prasetyo, "Chiller: Performance deterioration and maintenance," *Energy Engineering*, vol. 113, no. 4, pp. 55–80, 2016.

[10] "IoT communication sharing: Scenarios, algorithms and implementation," Tech. Rep., 2017, https://goo.gl/ogGbAB.

[11] J. Gao, J. Ploennigs, and M. Berges, "A data-driven meta-data inference framework for building automation systems," in *Proc. ACM Buildsys'15*, Seoul, South Korea, Nov. 2015.

[12] X. Wang, R. Habeeb, X. Ou, S. Amaravadi, J. Hatcliff, M. Mizuno, M. Neilsen, S. R. Rajagopalan, and S. Varadarajan, "Enhanced security of building automation systems through microkernel-based controller platforms," in *Proc. IEEE ICDCSW'17*, Atlanta, Georgia, USA, Jun. 2017.

[13] S. Latre, P. Leroux, T. Coenen, B. Braem, P. Ballon, and P. Demeester, "City of things: An integrated and multi-technology testbed for IoT smart city experiments," in *Proc. IEEE ISC2'16*, Trento, Italy, Sep. 2016.

[14] A. D'Alconzo, A. Coluccia, and P. Romirer-Maierhofer, "Distribution-based anomaly detection in 3G mobile networks: from theory to practice," *International Journal of Network Management*, vol. 20, no. 5, pp. 245–269, 2010.

[15] X. Wang, T. Kwon, Y. Choi, H. Wang, and J. Liu, "Cloud-assisted adaptive video streaming and social-aware video prefetching for mobile users," *IEEE wireless communications*, vol. 20, no. 3, pp. 72–79, 2013.

[16] S. C. Chan, K. Chan, K. Liu, and J. Y. Lee, "On queue length and link buffer size estimation in 3G/4G mobile data networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1298–1311, 2014.

[17] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[18] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, 2016.

[19] S. Park and Y. Yoo, "Network intelligence based on network state information for connected vehicles utilizing fog computing," *Mobile Information Systems*, vol. 2017, 2017.

[20] U. Brenner, "A faster polynomial algorithm for the unbalanced hitchcock transportation problem," *Operations Research Letters*, vol. 36, no. 4, pp. 408–413, 2008.

[21] C. Courcoubetis and R. Weber, *Pricing communication networks: economics, technology and modelling*. John Wiley & Sons, 2003.

[22] J. Huang and L. Gao, "Wireless network pricing," *Synthesis Lectures on Communication Networks*, vol. 6, no. 2, pp. 1–176, 2013.

[23] P. Slavík, "A tight analysis of the greedy algorithm for set cover," in *Proc. ACM STOC'96*, May. 1996.

[24] A. Libmodbus, "Modbus library for Linux, Mac OS X, FreeBSD, QNX and Win32. Libmodbus-a fast and portable Modbus library."

[25] T. Watteyne, K. Pister, D. Barthel, M. Dohler, and I. Auge-Blum, "Implementation of gradient routing in wireless sensor networks," in *Proc. IEEE GLOBECOM'09*, Dec. 2009.

[26] A. Ramadan, "District cooling designing for life power & cost saving," https://www.slideshare.net/AbdoRamadan1/district-cooling-design-case-study, accessed Jul, 2015.

[27] A. H. Beitelmal, C. Patel *et al.*, "A steady-state model for the design and optimization of a centralized cooling system," *International journal of energy research*, vol. 34, no. 14, pp. 1239–1248, 2010.

[28] H. Yoshida and S. Kumar, "ARX and AFMM model-based on-line real-time data base diagnosis of sudden fault in AHU of VAV system," *Energy Conversion and Management*, vol. 40, no. 11, pp. 1191–1206, 1999.