

Distributed Optimisation Framework for In-network Data Processing

Sepideh Nazemi, *Member, IEEE*, Kin K. Leung, *Fellow, IEEE*, and Ananthtam Swami, *Fellow, IEEE*

Abstract—In-Network Processing (INP) is an effective way to aggregate and process data from different sources and forward the aggregated data to other nodes for further processing until it reaches the end user. There is a trade-off between energy consumption for processing data and communication energy spent on transferring the data. Specifically, aggressive data aggregation consumes much energy for processing, but results in less data for transmission, thus using less energy for communications, and vice versa. An essential requirement in the INP process is to ensure that the user expectation of quality of information (QoI) is delivered during the process. Using wireless sensor networks for illustration and with the aim of minimising the total energy consumption of the system, we study and formulate the trade-off problem as a nonlinear optimisation problem where the goal is to determine the optimal data reduction rate, while satisfying the QoI required by the user. The formulated problem is a Signomial Programming (SP) problem, which is a non-convex optimisation problem and very hard to be solved directly. We propose two solution frameworks. First, we introduce an equivalent problem which is still SP and non-convex as the original one, but we prove that the strong duality property holds, and propose an efficient distributed algorithm to obtain the optimal data reduction rates, while delivering the required QoI. The second framework applies to the system with identical nodes and parameter settings. In such cases, we prove that the computational complexity of the problem can be reduced logarithmically such that each node can readily determine its optimal reduction rate locally. We evaluate our proposed frameworks under different parameter settings and illustrate the validity and performance of the proposed techniques through extensive simulation.

Index Terms—In-Network Processing, Quality of Information, Data aggregation, Distributed optimisation, Non-convex optimisation, Data reduction rate, Trade-off, Energy efficiency.

I. INTRODUCTION

IN an information network consisting of different types of communication devices equipped with various types of sensors, it is inevitable that a huge amount of data will be generated. Considering practical network constraints such as bandwidth and energy limitations, storing, processing and transmitting this very large volume of data are very challenging, if not impossible. However, In-Network Processing (INP) has opened a new door to possible solutions for optimising the utilisation of resources in such networks. INP aims to process and route data from different sources with the objective of reducing resource consumption, particularly energy [1].

While data aggregation helps optimise the usage of network

resources, it can also affect the quality of information (QoI) required by end user(s). Data aggregation can cause QoI to deteriorate [2]. The degree at which a system can process and aggregate information is one of the main determining factors of QoI. Despite the development of several data aggregation protocols, such as [3], [4], [5], [6], [7], [8] and [9] that ensure desired quality of service, data aggregation that guarantees required QoI has not been well studied. For this reason, we aim to investigate the QoI-aware INP problem.

Since data transmission is the main cause of energy consumption in the network, the idea of energy conservation via reducing the amount of data transmissions has caught the attention of many researchers. An early work on energy efficient data aggregation developed a data centric-routing scheme called direct diffusion [10]. In this work, if the attributes of data generated by the sources match the interest of the sink, a gradient specifying the data rate and the direction of transmission is set up to identify the data generated by the sensor nodes. Direct diffusion eliminates redundant transmissions by selecting only the relevant and useful data for transmission. Therefore, it can conserve a huge amount of energy. Moreover, cluster-based data aggregation protocols such as LEACH [11] and CLUDDA [12] have shown the effectiveness of this idea in prolonging network lifetime. Comprehensive reviews of data aggregation techniques can be found in [1], [13] and [14].

The common assumption among most of the data aggregation work is that the energy required to process data is less than the energy required to send it. Therefore, it is beneficial to perform computation to reduce the data volume for transmission. Very little attention has been paid to computational energy cost when data is aggregated in the network [15]. However, Barr and Asanovic [16] investigated energy saving by lossless data compression and showed that with several typical compression tools, there is a net energy increase when compression is applied before transmission. As for computational energy cost, Gallucio et al. [15] studied and estimated the conditions under which aggregation is preferable.

Eswaran et al. [17] formulated the INP problem as a concave optimisation problem to maximise a utility function of QoI. They applied the network utility maximisation (NUM) framework to determine the optimal compression and fusion factors for data aggregation as well as the optimal locations for performing data processing. The sum of energy consumption at each node was added to the objective function as a penalty term. They assumed that the energy consumption at each node is characterised by a convex function, which

Parts of the results in this paper were presented at IEEE Wireless Communications and Networking Conference, 2016 [31] and IEEE Personal, Indoor, and Mobile Radio Communications conferences, 2016 [32].

guarantees that the problem retains its convexity property. However, as we will discuss later, the energy consumption function is not always convex. Moreover, maximising the utility function of QoI may lead to waste of energy, if the user or application does not require the maximum utility of QoI.

With the aim of maximising aggregated information under a time constraint, Hariharan and Shroff [18] formulated a problem where the objective is to maximise the number of source nodes whose data are of interest to the sink. They proposed an algorithm that maximises the aggregated information in a given data aggregation tree when the sink imposes a deadline. Considering the QoI of data reported to the sink, [18] aims to implicitly maximise the QoI (i.e., aggregated data) by maximising the number of related source nodes. However, as we mentioned earlier, QoI is a subjective matter and may be varied in different applications. Maximising QoI in all circumstances without considering the user or application requirements can waste network resources including energy. Similarly, Alinia et al. [19] considered a deadline constraint to construct a data aggregation tree for maximal quality. They sought to maximise the number of participant nodes under the time constraint and proposed a near-optimal solution.

In our early work [20], we considered configuring a network with the same goal of minimising the total energy consumption as here and proposed a heuristic approach by imposing the same QoI constraint at each node, including the sink, to provide an approximate solution. The primary objective of this paper is to propose an exact optimal solution to the optimisation problem posed in [20]. Similarly, Sharma et al. [21] minimised the system energy consumption, while making a joint decision on whether and when to compress and transmit the data, by utilising the Lyapunov optimisation framework. The proposed framework only compresses information at sources which makes the rest of the problem a traditional scheduling or routing problem where a specific amount of data must be scheduled and transferred from the source to the sink. Data compression at intermediate nodes will raise the question of where and how to compress the data, making the scheduling and transmission decisions more challenging. It is especially so in the distributed scenario as each node needs to have information about how its neighbours are going to compress the data because it can affect not only their choice of compression, but also routing decisions and the optimality of the system energy consumption. It is very desirable for the network where nodes, regardless of their positions in the aggregation tree and just based on their characteristics and exchanges of a few control parameters, can efficiently and optimally compress the data in order to minimise the total energy consumption. A couple of numerical examples can be found in Section IV to illustrate this.

The problem of joint optimisation of processing, transmission and storage has recently been studied in the distributed cloud computing literature. For example, [22] addresses the problem of optimal placement of both content and virtual network function in a distributed cloud to minimise the overall network cost associated with storage, processing and transport of information subject to either

binary or fractional user flow. [22] formulated the problem as an integer linear programming problem and utilised the linear programming solver Xpress-MP to solve it. Furthermore, to minimise clouds operational cost, [23] tackled the problem of placement of services, routing of service flows and the associated allocation of cloud and network services where it is possible to compress or decompress data through network operations. The authors formulated the problem as a linear programming problem where the objective is to minimise the overall network resources cost as a function of integer numbers of allocated resources. Furthermore, [23] provided an approximate solution to the problem by proposing a queue-length based algorithm.

How INP should be carried out for satisfactory QoI at user level still remains an open issue. As an example, consider a system that monitors patients. The system must continuously gather data of interest for a long period. In such a system, in order to save communication energy consumption, medical data must be efficiently gathered and processed/aggregated while guaranteeing the quality of gathered data since the low quality of data (e.g., inaccurate, incomplete, false, etc.) may lead to human life risks [24]. In this system, minimising the cost of data communication by aggregation methods (e.g. compression, fusion, etc.) is desirable, it, however, will impact the quality of data. In this paper, we explore the trade-off among communication and computation energy and quality of data and concentrate on developing an analytical framework to facilitate controlling of INP at individual nodes along communication paths with consideration of the QoI requirement.

We formulate the problem of QoI-aware INP as a nonlinear optimisation problem to identify and control the optimal degree at which data can be aggregated at each node in the network in order to minimise the total energy consumption of the network subject to meeting the QoI requirement of the end user. Besides, in an inherently distributed, ad-hoc and dynamic environment such as WSNs with a large amount of data, distributed approaches to QoI-aware INP problems are very desirable in order to enhance reliability and adaptability of the network. Therefore, this work aims to develop a distributed framework that can effectively carry out INP for the satisfactory QoI required by the end user.

In common with prior work such as [17], [20] and [21], we consider computational cost. However, in contrast with the existing NUM work such as [17] where the concavity of the problem is a common assumption, which may not be valid in many applications [25], we formulate the problem as a non-convex signomial programming problem. We propose a solution and show the optimality of the solution by proving that the strong duality holds for the problem. By making use of the key continuity results in [25], we propose a novel distributed solution that efficiently achieves the optimal solution, despite strong inter-connectivity among variables and non-separability nature of the problem.

The main contributions of this paper are:

- Propose a novel formulation (i.e. SP problem) of QoI-aware INP problem.

- Prove the strong duality property under some cost assumptions and structures for the proposed non-convex SP optimisation problem.
- Devise a novel distributed optimisation framework based on tree structures for data aggregation.
- Develop an efficient and novel distributed algorithm based on the unique characteristic of the data aggregation tree (i.e., inter-dependency between a parent and its children) to achieve the global optimal solution for the QoI-aware INP problem.
- Evaluate and validate the correctness and performance of the proposed optimisation approaches for various scenarios and parameter settings.

The rest of the paper is organised as follows. We formulate the problem and define our network topology and assumptions, and discuss the challenges and hardness of the problem in section II. In Section II-D, we discuss challenges in solving the proposed problem and explain our proposed approaches to tackle this problem. In section III, we describe our proposed solutions in detail. We evaluate the correctness and performance of the proposed models through extensive simulation in section IV. Conclusions and future work are presented in section V.

II. PROBLEM FORMULATION

A. Assumptions

We assume that a data aggregation tree is formed among all involved nodes in the network after the user requests information from the network as illustrated in Fig. 1. We also assume that the root node, r , of the tree is responsible for delivering the required information to the end user. Without loss of generality, it is assumed that only leaf nodes generate data and each of the remaining nodes (intermediate nodes) in the tree receives data from its children, processes and forwards aggregated data to its parent and toward the root node, as shown in Fig.1b. We will show that this assumption (i.e., only leaf nodes generate data) can be relaxed.

We define the ratio of the volume of aggregated data to that of all data received at each node from its children as the *data reduction rate* denoted by $\delta \in (0, 1]$. The reduction rate is the degree to which a node can aggregate its received data efficiently, and a determining factor for QoI. We consider the amount of information that the user needs to receive (e.g., number of data packets) as a QoI requirement threshold. We acknowledge that besides the required amount of information, other parameters of QoI may also be important. The investigation of other QoI metrics and relaxation of assumptions will be left to our future work.

We also assume that data generated in the information networks has some redundancy due to the spatial and temporal correlation among sensor data. Therefore, it is possible to aggregate data as a means of reducing energy consumption for transmission and reception. However, the more the data is aggregated, the higher is the computational energy cost. Therefore, a trade-off exists among the energy that each node spends for data reception, transmission and computation.

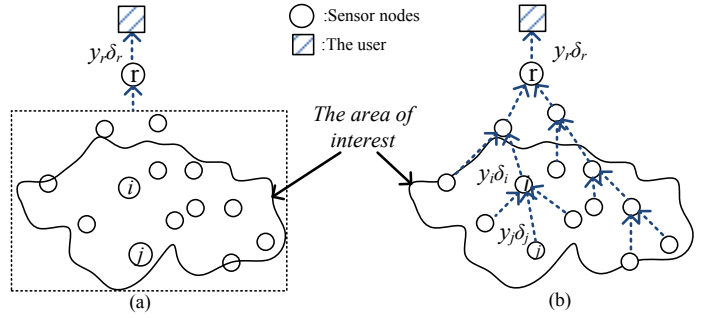


Fig. 1. (a) INP architecture; (b) Aggregation tree formed in responding to the query by the end user.

B. Problem Statement

We formulate the energy-efficient data aggregation problem subject to given QoI requirement as follows:

$$\begin{aligned} \min_{\delta} \quad & \sum_{i=1}^N F_i(\delta_i, y_i) \\ \text{s.t.} \quad & q_r(\delta_r, y_r) \geq \gamma \end{aligned} \quad (1)$$

where N is the total number of nodes in the data aggregation tree. $F_i(\delta_i, y_i)$ is the energy cost function of node i , which is a function of the total volume of input data y_i received from its children and the data reduction rate δ_i at node i . Since it is assumed that only leaf nodes generate data, we have

$$y_i = \sum_{j \in C_i} \delta_j y_j \quad \forall i \in \mathcal{N} - \mathcal{K}, \quad (2)$$

where \mathcal{N} and \mathcal{K} denote the set of all nodes and the set of all leaf nodes respectively. C_i denotes the set of children of node i . In addition, δ is a vector of reduction rates for all nodes. $q_r(\delta_r, y_r)$ specifies the QoI function. Since the root node, r , is responsible for delivering the required information to the end user, the QoI constraint is associated only with the root node. γ indicates the QoI requirement threshold specified by the end user. Even though the problem has only a single QoI constraint associated with the root node, the data reduction rate must be chosen optimally at every node so that the total energy consumption is minimised while the QoI constraint for the end user is satisfied.

Assuming the amount of data required by the end user to be the QoI requirement, i.e., $q_r(\delta_r, y_r) = y_r \delta_r$, the optimisation problem (1) becomes:

$$\begin{aligned} \min_{\delta} \quad & \sum_{i=1}^N F_i(\delta_i, y_i) \\ \text{s.t.} \quad & y_r \delta_r \geq \gamma \end{aligned} \quad (3)$$

where the constraint $y_r \delta_r \geq \gamma$ specifies the minimum volume of aggregated data that the user requires from all source (leaf) nodes in the area of interest, as shown in Fig. 1.

C. System Model

Let the total energy consumption of node i denoted by F_i consists of energy spent in receiving e_{iR} , computing e_{iC} and transmitting e_{iT} its data as follows:

$$F_i(\delta_i, y_i) = e_{iR} + e_{iC} + e_{iT}, \quad (4)$$

where,

$$e_{iR} = \epsilon_{iR} y_i, \quad (5)$$

$$e_{iT} = \epsilon_{iT} y_i \delta_i, \quad (6)$$

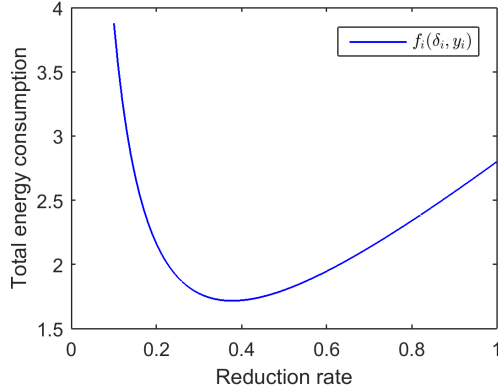


Fig. 2. Total energy consumption function associated to node i given assumptions (4)-(7), $l_i(\delta_i) = (\frac{1}{\delta_i} - 1)$, $\epsilon_C = 0.01$, $\epsilon_T = 0.05$, $\epsilon_R = 0.02$ and $y_i = 40$.

$$e_{iC} = \epsilon_{iC} y_i l_i(\delta_i), \quad (7)$$

and ϵ_{iR} , ϵ_{iC} and ϵ_{iT} are the energy consumed in receiving, processing and transmitting one unit of data, respectively, at node i . Since the greater the degree of data aggregation (i.e., the smaller amount of aggregated data produced after processing), the higher the energy consumption for computation, $l_i(\delta_i)$ is considered to be a scaling function to capture this characteristics of the computation energy consumption e_{iC} . In this work, we assume that $l_i(\delta_i)$ is a decreasing differentiable function of the reduction rate δ_i .

The intuition behind the definition of e_{iC} is that (i) computation cost is proportional to input data volume y_i as entire input data is expected to be scanned at least once before any computation, and (ii) typically, the greater the degree of data aggregation, the higher the energy consumption for computation. In addition, the influence of the reduction rate δ on e_{iC} is highly dependent on the type of data being aggregated, the actual aggregation functions and the characteristics of computation hardware [16]. For instance, processing high-quality video usually consumes more energy than processing data such as temperature or humidity measurements.

As an example, Fig.2 shows the total energy consumption of node i based on the assumptions in (4)-(7) with $l_i(\delta_i) = (\frac{1}{\delta_i} - 1)$ for $\delta_i > 0$. As illustrated, there is a trade-off between computation and transmission costs. At $\delta_i = 1$, node i sends all received data to its parent. That means $e_{iC} = 0$. A smaller δ_i means that node i reduces the volume of its transmitted data. Therefore, it can reduce its total energy consumption since it needs to transmit less data. However, after a certain point, aggregating more data will require spending more energy due to the need for more computation, even though it needs to spend less energy for transmission. Note that the node parameters ϵ_C and ϵ_T can affect this trend. Given (4)-(7), we define f_i as the total energy consumed by node i for **one unit** of received data:

$$f_i = \epsilon_{iR} + \epsilon_{iC} l_i(\delta_i) + \epsilon_{iT} \delta_i. \quad (8)$$

D. Problem Challenges and Solution Approaches

Following our system assumptions and cost model in (5) to (7), the problem (3) belongs to the category of non-convex problems called Signomial Programming (SP) problems [26]. Recall that in general, SP problems cannot be solved in polynomial time [26], [27].

The problem (3) is SP since, given (2), y_i is a posynomial function of all control variables associated with nodes located

at the sub-tree rooted at node i . Therefore, $F_i = y_i(\epsilon_{iR} + \epsilon_{iC} l_i(\delta_i) + \epsilon_{iT} \delta_i)$ is a signomial function. Since signomials are closed under addition, the problem (3) is a signomial programming problem. Furthermore, the assumption (2) leads to a high level of interdependency among nodes and makes the problem (3) inseparable. It is believed that the separability attribute of a problem can reduce computational complexity of the problem's solution considerably [28].

We convert the proposed problem into an equivalent problem which retains the non-convexity of the original problem. We show that strong duality holds for the equivalent problem; hence it can be solved exactly. We propose a novel strategy based on the gradient-based algorithm and design an iterative and effective algorithm that can achieve the optimal solution efficiently. Separately, we investigate conditions under which the complexity of the problem can be reduced and introduce a novel approach for complexity reduction of the proposed problem. We prove that under specific parameter settings, our model reduction technique can reduce the problem of N variables to an equivalent problem of $\log N$ variables. Consequently, the complexity of the optimisation problem will be reduced considerably.

III. PROPOSED SOLUTIONS

In this section, we propose two solution frameworks for solving the optimisation problem given in (3). In the first framework, we utilised the dual optimisation theory to develop a distributed solution to the problem. In the second one, we exploited the identical parameter settings to reduce the complexity of the solution approach significantly. These two frameworks are elaborated in the following subsections A and B, respectively.

A. Distributed Optimisation Framework

We transform the original non-convex problem into an equivalent, still non-convex, problem that is easy to solve.

Let $h(k)$ denote the depth of node k in the aggregation tree. The root node r is located at level 0 and its depth is 0. For any node k in the aggregation tree, let

$$\tau_k = (n_{k,h(k)}, n_{k,h(k)-1}, \dots, n_{k,1}, n_{k,0})$$

denote the unique path from node $n_{k,h(k)}$ to $n_{k,0}$ where by definition $n_{k,0} \triangleq 0$ (i.e., the root) and $n_{k,h(k)} \triangleq k$ (i.e., the node itself). Then, $n_{k,i}$ is the node at the i^{th} hop from the root in the unique path τ_k . We let $\delta_{k,i}$ and $f_{k,i}$ denote the corresponding reduction rate and energy consumption.

According to the assumptions in (4) to (7), the total energy consumption of each node i is directly proportional to the total volume of data y_i received from all its children where the proportionality constant is given by f_i in (8). Since the data received at each node is the sum of all output data from its children (see equation (2)), the linear relationship reveals that the total energy consumption at each node i is equal to the sum of energy spent on each of the data streams received from different children nodes. As only leaf nodes are assumed to generate data, we can reformulate the problem in (3) over all nodes in the network as one based on data generated at each

leaf node. Hence, the optimisation problem in (3) is equivalent to the following:

$$\begin{aligned} \min_{\delta} \quad & \sum_{k \in \mathcal{K}} y_k \left(f_k(\delta_k) + \sum_{i=0}^{h(k)-1} f_{k,i}(\delta_{k,i}) \prod_{m=i+1}^{h(k)} \delta_{k,m} \right) \\ \text{s.t.} \quad & \sum_{k \in \mathcal{K}} y_k \left(\prod_{i=0}^{h(k)} \delta_{k,i} \right) \geq \gamma \end{aligned} \quad (9)$$

Problem (9) retains the non-convexity of its original problem as a SP. However, it is more convenient for extracting the characteristics of the problem which may help us to solve the problem. Given that the KKT conditions are necessary conditions for the global optimal solution to the nonlinear primal problem in (9) [30], and by analysing the KKT conditions for the dual problem associated with (9), we prove in the following that the primal and dual problem have zero duality gap.

1) *Strong Duality Assessment:* Let $d(\lambda)$ be the Lagrangian dual function of (9) where λ is the Lagrangian multiplier (price) associated with the QoI constraint in (9). Then, the dual optimisation problem is as follows:

$$\max_{\lambda} d(\lambda) = L(\delta^*, \lambda). \quad (10)$$

The Lagrange function is given by

$$\begin{aligned} L(\delta, \lambda) = \quad & \sum_{k \in \mathcal{K}} y_k \left(f_k(\delta_k) + \sum_{i=0}^{h(k)-1} f_{k,i}(\delta_{k,i}) \prod_{m=i+1}^{h(k)} \delta_{k,m} \right) \\ & - \lambda \left(\sum_{k \in \mathcal{K}} y_k \left(\prod_{i=0}^{h(k)} \delta_{k,i} \right) - \gamma \right), \end{aligned} \quad (11)$$

and δ^* is obtained from

$$\delta^* = \underset{\delta}{\operatorname{argmin}} L(\delta, \lambda). \quad (12)$$

Let δ^* and λ^* be the primal and dual optimal solution. The KKT conditions stem from the fact that the gradient of the Lagrangian function must be zero [30]. That is

$$\nabla L(\delta^*, \lambda^*) = 0. \quad (13)$$

We consider the stationary condition for three different types of nodes namely, the root (node 0), the leaf (node k) and the intermediate node (node j) in the aggregation tree. The optimal data reduction rates δ_0 , δ_k and δ_j for these node types are obtained as follows.

Taking the partial derivatives of (11) at δ_0 , δ_k where $k \in \mathcal{K}$ and δ_j where $j \notin \mathcal{K}$, we have

$$\begin{aligned} \frac{\partial L}{\partial \delta_0} &= \sum_{k \in \mathcal{K}} y_k \left(f'_0(\delta_0) \prod_{m=1}^{h(k)} \delta_{k,m} \right) \\ & - \lambda \sum_{k \in \mathcal{K}} y_k \left(\prod_{m=1}^{h(k)} \delta_{k,m} \right) = 0. \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{\partial L}{\partial \delta_k} &= y_k f'_k(\delta_k) + y_k \sum_{i=0}^{h(k)-1} \left(f_{k,i}(\delta_{k,i}) \prod_{m=i+1}^{h(k)} \delta_{k,m} \right) \\ & - \lambda y_k \left(\prod_{m=0}^{h(k)-1} \delta_{k,m} \right) = 0 \quad \text{for } k \in \mathcal{K}. \end{aligned} \quad (15)$$

For ease of notation we let $\prod_{i=L}^U \delta_i = 1$, if $L > U$. Given that $\prod_{i \in \mathcal{N}} \delta_i \neq 0$, by factorising and rearranging the terms in

(14) and (15), we have

$$f'_0(\delta_0) = \lambda. \quad (16)$$

$$f'_k(\delta_k) = - \sum_{i=0}^{h(k)-1} \left(f_{k,i}(\delta_{k,i}) \prod_{m=i+1}^{h(k)-1} \delta_{k,m} \right) + \lambda \left(\prod_{m=0}^{h(k)-1} \delta_{k,m} \right). \quad (17)$$

To compute $\frac{\partial L}{\partial \delta_j}$, notice that δ_j only affects the energy cost of node j and its ancestors. Recall the definitions of τ_j and $h(j)$ and let y_j denote the total volume of incoming data at node j . Then

$$\begin{aligned} \frac{\partial L}{\partial \delta_j} &= y_j f'_j(\delta_j) + y_j \sum_{i=0}^{h(j)-1} f_{j,i}(\delta_{j,i}) \prod_{m=i+1}^{h(j)-1} \delta_{j,m} \\ & - \lambda y_j \prod_{m=0}^{h(j)-1} \delta_{j,m} = 0. \end{aligned} \quad (18)$$

We treat (18) in the same manner as (15) to obtain

$$f'_j(\delta_j) = - \sum_{i=0}^{h(j)-1} f_{j,i}(\delta_{j,i}) \prod_{m=i+1}^{h(j)-1} \delta_{j,m} + \lambda \prod_{m=0}^{h(j)-1} \delta_{j,m}. \quad (19)$$

Notice from (17) and (19) that we do not need to differentiate between the leaf nodes and intermediate nodes.

For the cost model in (8) with $l(\delta) = (1/\delta - 1)$, we have

$$f'_i(\delta_i) = \epsilon_{iT} - \frac{\epsilon_i C}{\delta_i^2}. \quad (20)$$

Substituting (20) into (16) and (19) lead to

$$\delta_0(\lambda) = \sqrt{\frac{\epsilon_0 C}{\epsilon_{0T} - \lambda}} \quad (21)$$

and

$$\delta_j(\lambda) = \sqrt{\frac{\epsilon_j C}{\epsilon_{jT} + \sum_{i=0}^{h(j)-1} (f_{j,i}(\delta_{j,i}) \Delta_{j,i}) - \lambda(\Delta_j)}}. \quad (22)$$

where $\Delta_{j,i} = \prod_{m=i+1}^{h(j)-1} \delta_{j,m}$ and $\Delta_j = \prod_{m=0}^{h(j)-1} \delta_{j,m}$.

Equations (21) and (22) are called the price-based solution functions for the problem (9), because they are expressed as functions of the Lagrangian multiplier (price) λ . Equations (21) and (22) are for the specific cost function (8). For a general cost function which is monotone decreasing and differentiable with invertible derivative, δ can be uniquely determined from $f'(\delta)$. The computation of the optimal λ is discussed later in this section. By applying the key results in [25], we have the following Theorem.

Theorem 1. *The strong duality property holds for the primal-dual problem in (9) and (10). Furthermore, an iterative algorithm exists to obtain the optimal solution for both problems.*

Proof. Observe from (21) and (22) that the optimal data reduction rates for all nodes in the aggregation tree are continuous functions of the price variable λ in the feasible range, including the optimal value of λ^* . Based on Theorem 1 in [25], this continuity property guarantees that the duality gap for (9) and (10) is zero and that the optimal solutions for the primal-dual problems can be obtained by an iterative method. \square

2) *Relaxing assumptions on data generation:* We have assumed above (see equation (2)) that only leaf nodes can generate data and intermediate nodes only process received data

from their children and transmit the aggregated data. In the following, we show that (21) and (22) are still valid if we relax the assumption (2). That means in addition to the leaf nodes, the intermediate nodes can generate or sense data as well.

Let the amount of data at the intermediate node i be

$$y_i = \overset{\circ}{y}_i + \sum_{j \in C_i} \delta_j y_j, \quad \forall i \in \mathcal{N} - \mathcal{K}, \quad (23)$$

where $\overset{\circ}{y}_i$ is the amount of data generated (sensed) locally by node i , and as defined previously, $\sum_{j \in C_i} \delta_j y_j$ denotes the amount of data that node i receives from its children.

Since the total energy consumption of node i is still directly proportional to the total volume of data held by node i and given the additive attribute of the model (23), problem (9) becomes

$$\begin{aligned} \min_{\delta} \quad & \sum_{k \in \mathcal{K}} y_k \left(f_k(\delta_k) + \sum_{i=0}^{h(k)-1} f_{k,i}(\delta_{k,i}) \prod_{m=i+1}^{h(k)} \delta_{k,m} \right) \\ & + \sum_{i \notin \mathcal{K}} \overset{\circ}{y}_i \left(f_i(\delta_i) + \sum_{z=0}^{h(i)-1} f_{i,z}(\delta_{i,z}) \prod_{m=z+1}^{h(i)} \delta_{i,m} \right) \end{aligned} \quad (24)$$

s.t.

$$\sum_{k \in \mathcal{K}} y_k \left(\prod_{i=0}^{h(k)} \delta_{k,i} \right) + \sum_{i \notin \mathcal{K}} \overset{\circ}{y}_i \left(\prod_{m=0}^{h(i)} \delta_{i,m} \right) \geq \gamma.$$

We treat (24) in the same manner as (9) in order to obtain price-based solution functions. That means taking the partial derivative of the Lagrange function corresponding to the problem (24) at δ_0 (the root), k (the leaf node) and j (the intermediate node). Details of the derivation are provided in Appendix A. As a result, price-based functions form in both cases (i.e., (i) only the leaf nodes generate data and (ii) all nodes generate data) are identical.

3) *Distributed Algorithm*: The gradient-based method is a popular technique to find *local optima*. At each iteration, the search continues in the positive direction of the gradient of the function. The gradient-based recursion for solving (10) is given by:

$$\lambda^{(t+1)} = \lambda^{(t)} + \alpha \left(\frac{\partial L(\delta^*(\lambda), \lambda)}{\partial \lambda} \right), \quad (25)$$

where t is the iteration index and $\alpha \geq 0$ is the step size. However, It is important to note that since all price-based solution functions, namely (21) and (22), associated with problem (9) are continuous over λ 's domain, we can devise an iterative algorithm based on (25) that converge to the *global optimal* solution, as suggested by Theorem 1.

Due to the complex relationships among nodes imposed by the tree structure, and the assumption in (2), the problem (9) and its corresponding Lagrangian function (11) cannot be easily separated to develop a distributed solution. However, a careful observation of (21) and (22) reveals that the optimal data reduction at a node (say node i) only depends on the optimal reduction rates of all ancestors of node i in the aggregation tree and the optimal price value λ^* . That is, if a node knows just the solutions associated with its ancestors and the optimal price value λ^* , it can calculate its data reduction rate δ . We exploit this critical observation to devise our distributed solution as follows.

Given λ (not necessarily the optimal one), the root node can compute $\delta_0(\lambda)$ by inverting (16); for the specific choice of ℓ , this is obtained via (21). The root node can then compute

Algorithm 1: Distributed calculation of data reduction rates

input : λ
output: $\delta_j, \forall j \in \mathcal{N}$

- 1 **Root node:**
- 2 Computes δ_0 from (16) or (21);
- 3 Computes $s = f_0(\delta_0)$, $w = \lambda \delta_0$;
- 4 **Each node j , upon receiving (s, w) from its parent:**
- 5 Computes $f'(\delta) = -s + w$ and solves for δ from (19) or (22);
- 6 **If not a leaf node:**
- 7 Compute $s = s\delta + f(\delta)$, $w = w\delta$;
- 8 Transmit (s, w) to its children;
- 9 **end**

Algorithm 2: updating λ

- 1 **Each leaf node, $k \in \mathcal{K}$:**
- 2 Computes $v_k = y_k \delta_k$;
- 3 Transmits v_k to its parent;
- 4 **Each non-leaf node, j , upon reception of v_k from all its children:**
- 5 Computes $v_j = \delta_j \sum_{i \in C_j} v_i$;
- 6 **If j is not the root:**
- 7 Transmit v_j to its parent;
- 8 **else**
- 9 Update λ via (25);
- 10 **end**

$s = \delta_0 f_0(\delta_0)$ and $w = \lambda \delta_0$ and send (s, w) to its children. From (19), we see that when $h(j) = 1$, these values suffice to compute $f'(\delta_j)$, which can then be inverted to obtain δ_j ; (22) does so for the assumed $\ell(\delta)$. These nodes at level 1 of the tree then compute $s\delta_j + f(\delta_j)$ and $w\delta_j$ and pass these to their children. The recursion obviously stops at the leaves. At the end of this step, all nodes in the tree have computed δ_j according to (16) and (19), or (21) and (22) for the assumed ϵ_{jC} and ϵ_{jT} . This is illustrated in Algorithm 1.

The update of λ via (25) requires computation of the left hand term in (26). Each leaf node knows its own y_k and δ_k ; so each leaf node can transmit $v_k = y_k \delta_k$ to its parent. When a parent node, say j , has received the v_k values from all its children it computes $v = \delta_j \sum_{i \in C_j} v_i$ and transmits it to its parent. The recursion stops at the root node. This is illustrated in Algorithm 2.

The root node can then update λ , check to see if convergence has been achieved; if not, it restarts Algorithm 1.

4) *Solution Feasibility*: Consider the constraint in the problem (9). That is,

$$\sum_{k \in \mathcal{K}} y_k \left(\prod_{i=0}^{h(k)} \delta_{k,i} \right) \geq \gamma. \quad (26)$$

By assumption, $0 < \prod_{i=0}^{h(k)} \delta_{k,i} \leq 1, \forall k \in \mathcal{K}$. Therefore, if $\sum_{k \in \mathcal{K}} y_k \leq \gamma$, a feasible solution does not exist for the problem (9). Obviously, a feasible solution can exist only if the total amount of data generated in the network is greater than or equal to the QoI requirement.

Moreover, by definition we have $0 < \delta_i \leq 1$. Therefore, if $\delta_i, \forall i \in N$ determined during iterations of Algorithm 1 is outside of this box constraint (i.e., $\delta_i^*(\lambda^*) < 0, \delta_i^*(\lambda^*) > 1$ or δ_j is imaginary), we map the solution to the upper bound of data reduction rate value (i.e., $\delta_i^*(\lambda^*) \mapsto 1$). The intuition behind this is explained as follow.

From (16) and (20) we have:

$$f'_0(\delta_0) = -\epsilon_T - \frac{\epsilon_C}{\delta_0^2} = \lambda. \quad (27)$$

Note that $f'_0(\delta_0)$ is an increasing function of δ_0 . During iteration of Algorithm 2, we update the value of λ . The maximum value of λ in this case occurs when δ_0 reaches the upper bound boundary (i.e., 1). Therefore if the optimal value of λ causes δ_0 to attain an imaginary value, we map the solution to the upper bound value. In addition, if $\lambda^* = 0$ and $\epsilon_{0C} \gg \epsilon_{0T}$ then $\delta_0(\lambda) > 1$. In this case, we map the solution to the upper bound as well (i.e., $\delta_0^*(\lambda^*) \mapsto 1$) since, due to the high cost of computation, sending all information (choosing $\delta_0 = 1$) is more energy efficient than processing and reducing the data. Note that the same argument is applicable at the intermediate or leaf nodes.

Here ends the discussion of the distributed solution to the optimisation problem in (3). In the following subsection, we show that if all nodes have identical parameters, the complexity of the solution can be reduced drastically.

5) *Time Complexity*: To analyse the time complexity of the proposed solution, we consider a perfect c -ary aggregation tree. In this case, the time complexity of Algorithm 1 is $c \log_c N$, where N, c and $\log_c N$ are the total number of nodes, degree and depth of the aggregation tree, respectively. Note that the summation statement in line 5 of the algorithm must be executed c times for each parent. The time complexity of Algorithm 2 is $\log_c N$ as leaf nodes must calculate and send their QoI contributions to their parent nodes until their QoI contributions reach the root. Since these two algorithms run one after each other, the time complexity of running both Algorithms 1 and 2 once is $(c + 1) \log_c N$. Let X be the total number of iterations before convergence to the optimal solution. Then, the time complexity for reaching the optimal solution is $X(c+1) \log_c N$. In evaluating our solution technique for different parameter settings, we have found that the value of X is in the order of 10 in all cases considered.

B. Balanced tree with identical nodes

We prove in the following that under the assumption of identical parameter settings, the processing tree can be collapsed to a linear graph where the number of nodes represents the node levels of the original processing tree, thus significantly reducing the computation complexity. Toward this end, we introduce the following Lemma and Theorem.

Lemma. *Assume a perfect c -ary aggregation tree with depth h where: (a) apart from the leaf nodes, each node has exactly c children, (b) every leaf node generates the same amount of data, while all other nodes only receive data from their children, process the received data and transfer the aggregated data to one's parent, and (c) the network and energy consumption parameters in (5) to (7) are identical for*

all nodes. Under these assumptions, all the nodes at the same level of the tree (say level i) have an identical data reduction rate δ_i in the optimal solution for the problem (3).

Proof. See Appendix B. \square

The following Theorem shows that under the assumptions made in the Lemma, the optimal solution for the problem (3) associated with a symmetric aggregation tree is identical to that for a linear graph of $h+1$ nodes. For example, the optimal solution for the proposed problem for the tree in Fig.3a is identical to that for the linear graph in Fig.3b which represents a very significant reduction in complexity.

Theorem 2. *For the perfect c -ary aggregation tree with h levels and the assumptions made in the Lemma, the optimal reduction rates for all nodes at the same tree level for the problem (3) are identical to that of the node at the corresponding level in a linear graph with $h+1$ nodes. Furthermore, the optimal rates can be obtained by solving the following problem of the linear graph with $h+1$ nodes:*

$$\begin{aligned} \min_{\delta} \quad & Y \sum_{j=0}^h \left(\prod_{k=j+1}^h \delta_k \right) f(\delta_j) \\ \text{s.t.} \quad & Y \prod_{k=0}^h \delta_k \geq \gamma \end{aligned}, \quad (28)$$

where $\delta_{h+1} = 1$, $f(\delta_j) = \epsilon_R + \epsilon_T \delta_j + \epsilon_C l(\delta_j)$, h is the number of levels in the aggregation tree which is equal to $\log_c N$, δ is a vector of reduction rates associated with each level of the tree and γ is the amount of data generated at each leaf, $Y = \gamma c^h$ is the total amount of data generated at the leaf nodes.

Proof. At level i , there are c^i nodes. Each node j , where $j \in \{c^i\}$, receives a volume of data equal to

$$y_j = \gamma c^{h-i} \prod_{k=i+1}^{h+1} \delta_k, \quad \text{for } i = 1 \text{ to } h, \quad (29)$$

where $\delta_{h+1} = 1$.

The energy cost per unit of data at node j is equal to $\epsilon_R + \epsilon_T \delta_j + \epsilon_C l(\delta_j)$. Therefore, the total cost at level i is:

$$\begin{aligned} \phi_i(\delta_i, y) &= c^i \left(\gamma c^{h-i} \prod_{k=i+1}^{h+1} \delta_k \right) (\epsilon_R + \epsilon_T \delta_i + \epsilon_C l(\delta_i)) \\ &= \gamma c^h \left(\prod_{k=i+1}^{h+1} \delta_k \right) f(\delta_i). \end{aligned} \quad (30)$$

So, the total cost is

$$F(\delta_i, y) = \gamma c^h \sum_{i=0}^h \left(\prod_{k=i+1}^{h+1} \delta_k \right) f(\delta_i). \quad (31)$$

Since the root r is located at level 0, the amount of data received by the root is equal to

$$Y_r = \gamma c^h \prod_{k=1}^h \delta_k. \quad (32)$$

Therefore, the constraint in problem (3) can be written as

$$\gamma c^h \left(\prod_{k=1}^h \delta_k \right) \delta_0 = \gamma c^h \left(\prod_{k=0}^h \delta_k \right) \geq \gamma. \quad (33)$$

\square

In general, the complexity of an optimisation problem

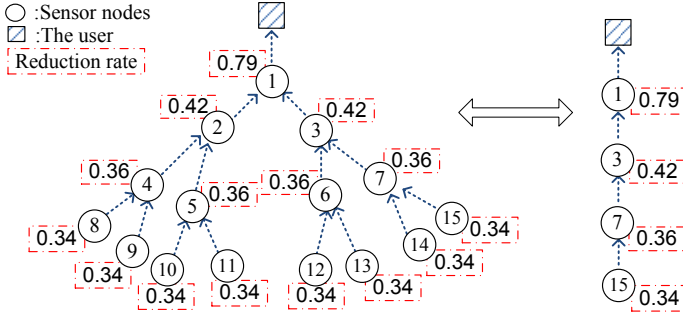


Fig. 3. (a) Reduction rates for nodes in a symmetric aggregation tree. (b) Network topology after model reduction.

solver depends on the number of variables in the problem [30]. Since our model reduction technique can reduce the problem of N variables to the equivalent problem of $\log_c N + 1$ variables, the complexity of the optimisation solver will be reduced considerably.

The significance of this proposed solution for the global optimisation problem is explained as follows. In general, the problem of in-network processing is a non-convex optimisation problem, for which no exact distributed solution technique has been found. Solving the problem by a centralized method does not scale in terms of algorithm complexity and incurs significant overhead in distributed environments. Under the assumption of identical parameters for all nodes, we have proved in Theorem 2 that the optimal data reduction rates for a symmetrical tree of N nodes can be obtained by solving the corresponding problem of a linear graph of $\log_c(N) + 1$ nodes. This represents a very significant reduction in complexity such that each node can obtain its optimal reduction rate by solving the much reduced problem locally.

In fact, Theorem 2 can be extended to the aggregation tree where parameters for nodes at the same tree level are identical but can vary from level to level. In that case, the optimal data reduction will be the same for all nodes at a given level and are identical to that of the node at corresponding level in the equivalent linear graph.

IV. PERFORMANCE EVALUATION

In this section, we present numerical results to validate the proposed approaches. In the first experiment, we consider 15 homogeneous nodes forming a 4-level perfect binary aggregation tree with parameters $\epsilon_C = 0.01, \epsilon_T = \epsilon_R = 0.02$ for each node. This way, we can compare results of the proposed distributed method with the optimal solution generated by exhaustive search. We also utilized sequential quadratic programming (SQP) method in the Matlab optimisation tools to obtain the the optimal solution for the problem (3). The function $l_i(\delta_i) = \frac{1}{\delta_i} - 1$, for $\delta_i > 0$ is considered as the scaling function. Each leaf generates $y = 15$ packets and the user's QoI requirement is considered to be $\gamma = 5$. Given these parameter settings, the existence of a feasible solution is guaranteed and it is in the interior. Note that a feasible solution will not exist if sum of the data generated at leaf nodes is less than the threshold γ .

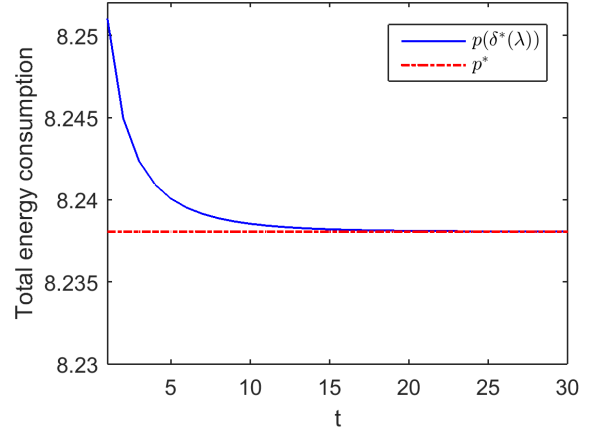


Fig. 4. Convergence of the proposed method vs. iteration index t .

Fig.3a shows the optimal data reduction rate at each node. As stated in the Lemma, under identical parameter settings, the optimal reduction rates for all nodes at the same level of the tree are identical. Total energy consumption of the network is 8.23 units and the root (node 1) delivers exactly 5 packets. Fig.3b illustrates the same network after applying our model as specified in the Theorem 2. It can be seen that the optimal reduction rates are identical to those in the original network. Note that the results provided in Fig.3 also have been confirmed by an exhaustive search algorithm.

Under the same parameter setting, in the next experiment, we aim to evaluate the proposed distributed optimisation framework in Section III-A.

Fig.4 presents the convergence of the proposed distributed optimisation algorithm versus the number of iterations. The solid line presents the value of the objective function $p(\delta(\lambda))$ at each iteration and the dashed line determines the optimal value of the objective function denoted by p^* and obtained by an exhaustive search algorithm. As the graph shows, after 20 iterations or so, the distributed algorithm converges to the optimal solution when the residual inconsistency value equals zero. That is, $\frac{\partial L(\delta, \lambda)}{\partial \lambda} = 0$, where the price value gains its optimal value 0.004141. In this experiment, the initial value of the Lagrangian multiplier λ and the step size α were set at 0 and 0.001 respectively.

Since we consider a symmetrical aggregation tree with homogeneous nodes, nodes located at the same level of the tree have identical optimal data reduction rates (as proved in the Lemma in section III-B). Fig.5 illustrates the variation of the reduction rates for nodes at level 0, 1, 2 and 3 after each iteration. As illustrated in Fig.5, while low level nodes have slight variation in adjusting their optimal data reduction rates, top level nodes need larger adjustment in their reduction rate specially during first couple of iterations. Fig. 6 shows the convergence of the price value to the optimum value $\lambda^* = 0.004141$ versus each iteration of algorithm.

To further illustrate and evaluate the effectiveness of our

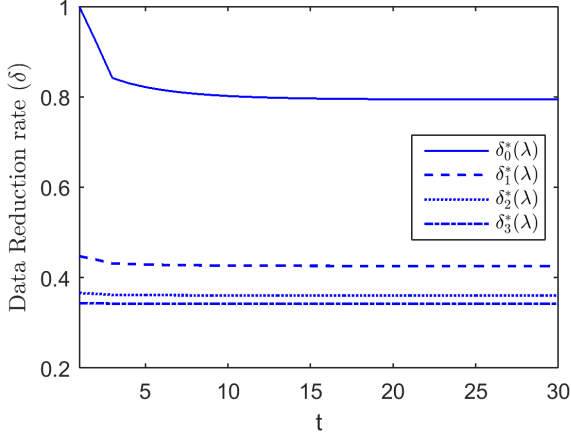


Fig. 5. Reduction rate's alteration vs. iteration index t for nodes at different levels of the tree.

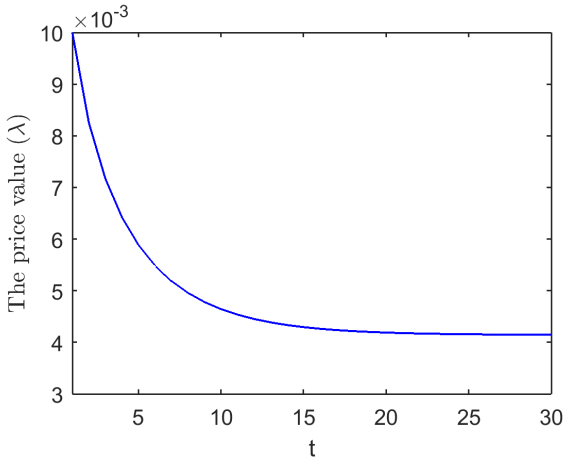


Fig. 6. The price value λ vs. iteration index t

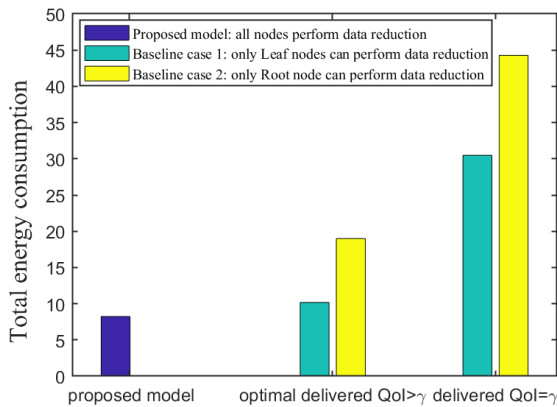


Fig. 7. Effectiveness of the proposed framework vs. two baseline cases

proposed framework and distributed algorithm, we compare our solution to two baseline cases (referred to as baseline case 1 and 2 below) where data can be compressed (reduced) only at the leaf (source) and root (destination) nodes, respectively.

TABLE I
PARAMETER SETTINGS CORRESPONDING TO FIG. 8

$\frac{\epsilon_C}{\epsilon_T}$	λ_0	α
50	0.0002	10^{-7}
5	0.0002	10^{-7}
1	0.0002	10^{-7}
0.5	0.01	0.001
0.2	0.02	0.001
0.02	0.1	0.01

As discussed above, our problem formulation allows every node in the aggregation tree to possibly perform data reduction. However, baseline case 1 corresponds to the scenario considered in [21] where only the source node can reduce the data volume before transmission. On the other hand, baseline case 2 allows data to be compressed only at the destination node, just to represent another extreme situation. For a given network and parameter setting as stated at the beginning of this section, our proposed solution technique is applied to identify the optimal data reduction rates at all nodes in order to minimise the overall energy consumption, which turns out to be 8.23, as shown on the far left-hand side of Fig. 7. As for the baseline cases, the only decision variable is the data reduction rate at the source or root nodes in order to meet or exceed the QoI threshold $\gamma = 5$. When the delivered QoI equals γ , the energy consumption for the two baseline cases of 30.48 and 44.27 respectively, are presented on the right-hand side of Fig. 7. To consider the possibility of lowering the total energy consumption when the delivered QoI is greater than γ , we performed an exhaustive search for the optimal reduction rate at the source and root nodes for the baseline cases. The minimum energy consumption, which is determined to be 10.18 and 18.99 for both respective baseline cases, is labelled as "optimal delivered QoI $> \gamma$ " at the middle of Fig. 7. The corresponding delivered QoI for baseline case 1 and 2 are 32 and 84, respectively, while the delivered QoI by our proposed framework is 5, which is precisely the required QoI level. By considering the total energy consumption, our proposed framework consumes much less energy than both baseline cases, while meeting the QoI requirement. Clearly, these results highlight the advantage of possible data compression on every node in the aggregation tree, as our proposed framework allows, in terms of minimising total energy consumption with satisfied QoI.

We evaluate the performance and correctness of the distributed algorithm under different parameter settings. In particular, we consider the computation energy ϵ_C to transmission energy ϵ_T ratio and test the algorithm under various values from extreme to moderate cases as presented by Table I. The optimal reduction rate at each level associated with these parameters settings is illustrated in Fig. 8. According to Fig. 8, when ϵ_C is much greater than ϵ_T (an extreme case), all nodes at each level will send all received data. That means they will not compress data due to extremely high cost of computation. In contrast, if $\epsilon_C \ll \epsilon_T$, it is beneficial to compress the data at lower levels of the aggregation tree in order to spend less

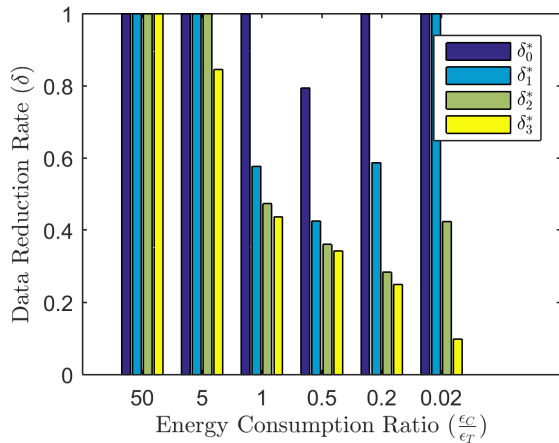


Fig. 8. Computation (e_C) and Transmission (e_T) energy trade-off at each level vs. iteration index t

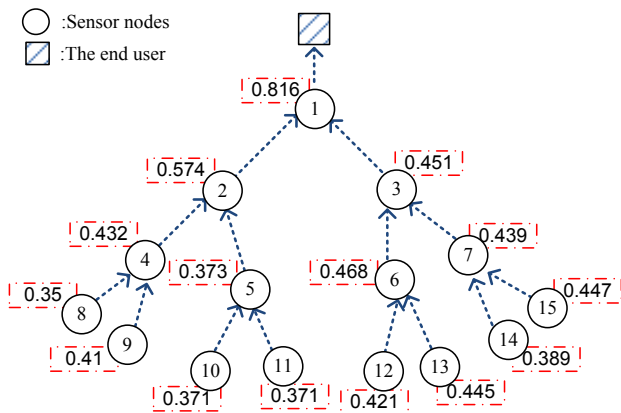


Fig. 9. Data reduction rates corresponding to a tree with heterogeneous nodes (i.e., various ϵ_C , ϵ_T and ϵ_R values at each node). $\lambda^* = 0$, delivered $QoI = 8$

energy for transmitting data. The increasing trend of data reduction rate from leaf nodes to the root node continues in other moderate parameter settings as illustrated in Fig. 8. However, this trend is only valid for a symmetrical data aggregation tree. A different pattern can be observed when we consider an irregular data aggregation tree with heterogeneous nodes as illustrated in Fig. 9. In this experiment, ϵ_C is randomly chosen from the interval $[0.01, 0.02]$. Furthermore, it is assumed that $\epsilon_T = \epsilon_R = 0.02$. The input data at each leaf node is 15 packets. The algorithm converged after 560 iterations and the delivered QoI and optimal price are 8 and 0 respectively.

Table II presents more experimental results regarding Fig. 8. It can be seen that the optimal price values attain a positive value when the delivered QoI is exactly equal to QoI threshold γ . That means the constraint in (9) is active, $\sum_{k \in \mathcal{K}} y_k (\prod_{i=0}^{h(k)} \delta_{k,i}^*) = \gamma$. On the other hand, when the delivered QoI is greater than γ , the Lagrangian multiplier (price) equals zero. This result is compatible with the fact that for any primal and dual optimisation problem with zero duality gap, the complementary slackness condition of KKT conditions must be satisfied by both the optimal primal solution and the

TABLE II
THE RESULTS ASSOCIATED WITH EXAMINING THE DISTRIBUTED ALGORITHM UNDER DIFFERENT PARAMETERS' SETTINGS

$\frac{\epsilon_C}{\epsilon_T}$	λ^*	Delivered QoI	Num. of Iteration
50	0	120	17
5	0	101	20
1	0	14	212
0.5	0.0041	5	324
0.2	0.121	5	112
0.02	2.444	5	72

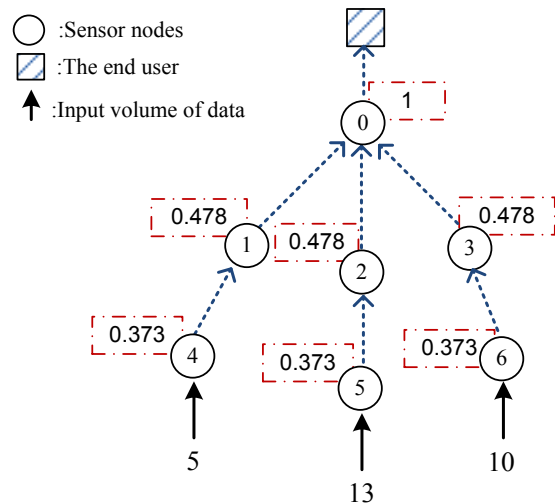


Fig. 10. Data reduction rate corresponding to the aggregation tree with 7 homogeneous nodes with different volume of data generated at each leaf .

optimal dual solution (the optimal Lagrangian multiplier). That means $-\lambda^* (\sum_{k \in \mathcal{K}} y_k (\prod_{i=0}^{h(k)} \delta_{k,i}^*) + \gamma) = 0$. Furthermore, Table II presents the number of iterations required for convergence. Note that initial price value of λ can affect the speed of convergence (compare this result to that in the previous experiment).

In the next experiment, in contrast to considering the complete binary data aggregation tree, we examine our model on various types of tree structure. We choose a small random tree with 7 identical nodes; however, the volume of data generated at each leaf node is different as illustrated in Fig. 10. In this experiment, $\epsilon_C = 0.01$ and $\epsilon_T = \epsilon_R = 0.02$, the QoI requirement $\gamma = 5$. Based on Fig. 10, given identical parameter setting for all nodes (i.e., $\epsilon_C, \epsilon_T, \epsilon_R$), data reduction associated to the nodes at the same level of data aggregation is identical. The optimal price value is $\lambda^* = 0.016$ and delivered QoI equals 5. These results were also confirmed by an exhaustive search. We repeated the same experiment over a larger random tree topology. According to Fig. 11, the leaf nodes have identical data reduction rate despite being located at different depths of the tree or having different input data.

These results indicate that the proposed model treats nodes fairly despite having variant input data. Considering that data from all the nodes have the same priority, the proposed model obtains the optimal data reduction rate for all nodes only based on node's characterization. However, if data generated at the nodes have different priorities, one can accommodate this

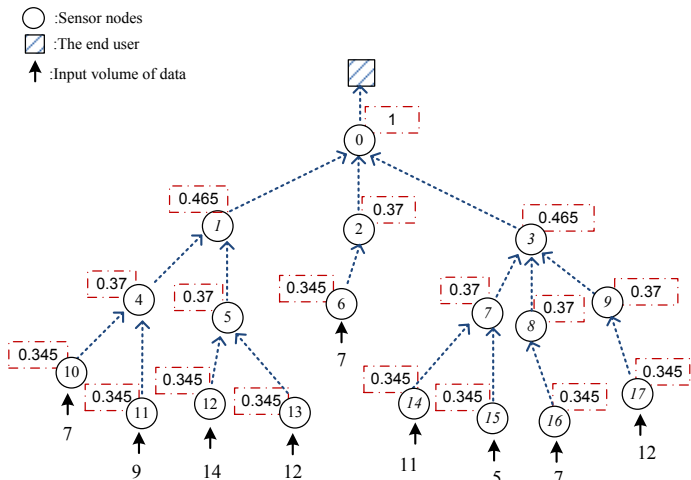


Fig. 11. Data reduction rate corresponding to the random aggregation tree with 17 homogeneous nodes and with different volume of data generated at each leaf .

priority into the computational cost model. For instance, data generated close to the area of interest or monitored event may have higher importance, since they may have more precise information than other information generated in the network. Data with high priority or sensitive information need more sophisticated processing algorithm which may lead to higher computation energy consumption than data with no priorities.

V. CONCLUSION

We studied and formulated a QoI-aware INP problem as a non-convex problem and established a framework that optimises the trade-off among computation and communication energy cost and the QoI required by the end user. We proposed a transformation of the original problem to an equivalent problem which still retains the non-convexity of the original problem. We proved that the new problem could be solved exactly since we show that the associated optimal duality gap is zero. We have devised a distributed gradient-base algorithm and showed that the method achieves the optimal solution efficiently. Furthermore, we have proved that under the assumption of identical parameter setting for nodes at the same level in the aggregation tree, the computation complexity of the problem can be reduced significantly (i.e., by the logarithmic scale). We evaluated our proposed frameworks under different parameter settings and illustrated the performance of our proposed models through extensive simulation and comparison with baseline approaches.

Our future work includes consideration of multiple users with different QoI requirements as overlay of the corresponding aggregation trees over a given physical network as well as the extension of the complexity reduction framework to heterogeneous node settings and characterisation of general network conditions while keeping the problem solvable.

ACKNOWLEDGMENT

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions

contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. Furthermore, the authors would like to thank Dr. Mahsa Derakhshani of Loughborough University for many helpful discussions and comments.

REFERENCES

- [1] Elena Fasolo, Michele Rossi, Jorg Widmer and Michele Zorzi, "In-network aggregation techniques for wireless sensor networks: a survey", *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 70–87, 2007.
- [2] Sylvanus A. Ehikioya, "A characterization of information quality using fuzzy logic", *IEEE International Conference on Fuzzy Information Processing Society*, pp. 635–639, 1999.
- [3] Hamed Yousefi, Marzieh Malekimajd, Majid Ashouri and Ali Movaghar, "Fast aggregation scheduling in wireless sensor networks", *IEEE Transactions on Wireless Communications*, vol.14, no. 6, pp. 3402–3414, 2015.
- [4] Yang Yu, Bhaskar Krishnamachari, and Viktor K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks", *IEEE International Conference on Computer Communications (INFOCOM)*, vol. 1, 2004.
- [5] Bo Yu, Jianzhong Li and Yingshu Li, "Distributed data aggregation scheduling in wireless sensor networks", *IEEE International Conference on Computer Communications (INFOCOM)*, pp. 2159–2167, 2009.
- [6] Yao Lu, Ioan Sorin Comsa, Pierre Kuonen, and Beat Hirsbrunner, "Dynamic data aggregation protocol based on multiple objective tree in Wireless Sensor Networks", *IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2015.
- [7] Imane Horiya Brahmi, Soufiene Djahel, Damien Magoni and John Murphy, "A spatial correlation aware scheme for efficient data aggregation in Wireless Sensor Networks", *Conference on Local Computer Networks Workshops (LCN Workshops)*, pp. 847–854, 2015.
- [8] Zhang Jiao, Ren Fengyuan, He Tao and Lin Chuang, "Data aggregation protocol based on dynamic routing in wireless sensor networks", *Conference on Communications and Mobile Computing (CMC)*, vol. 1, pp. 501–505, 2009.
- [9] Tian He, Brian M Blum, John A Stankovic and Tarek Abdelzaher, "AIDA: Adaptive application-independent data aggregation in wireless sensor networks", *Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 2, pp. 426–457, 2004.
- [10] Chalermek Intanagonwiwat, Ramesh Govindan and Deborah Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks", *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 56–67, 2000.
- [11] Anantha P. Chandrakasan and Wendi Beth Heinzelman, "Application-specific protocol architectures for wireless networks", *IEEE Transactions on Wireless Communications*, 2000.
- [12] Supriyo Chatterjea and Paul Havinga, "A dynamic data aggregation scheme for wireless sensor networks", *Program for Research on Integrated Systems and Circuits, Veldhoven, The Netherlands*, 2003.
- [13] Ramesh Rajagopalan and Pramod K Varshney, "Data aggregation techniques in sensor networks: A survey", <http://surface.syr.edu/eecs/22>, 2006.
- [14] Paulo Jesus, Carlos Baquero and Paulo Sérgio Almeida, "A survey of distributed data aggregation algorithms", *IEEE Communications Surveys & Tutorials*, vol.17, no. 1, pp. 381–404, 2015.
- [15] Laura Galluccio, Sergio Palazzo and Andrew T. Campbell, "Efficient data aggregation in wireless sensor networks: an entropy-driven analysis", *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2008.
- [16] Kenneth C. Barr and Krste Asanović, "Energy-aware lossless data compression", *ACM Transactions on Computer Systems (TOCS)*, vol. 24, no. 3, pp. 250–291, 2006.

- [17] Sharanya Eswaran, James Edwards, Archan Misra, and Thomas F. Laporita, "Adaptive In-Network Processing for Bandwidth and Energy Constrained Mission-Oriented Multihop Wireless Networks", *IEEE Transactions on Mobile Computing*, vol. 11, no. 9, pp. 1484–1498, 2012.
- [18] Srikanth Hariharan, Ness B Shroff, "Maximizing aggregated information in sensor networks under deadline constraints", *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2369–2380, 2011.
- [19] Bahram Alinia, Mohammad H Hajiesmaili and Ahmad Khonsari, "On the construction of maximum-quality aggregation trees in deadline-constrained WSNs", *IEEE Conference on Computer Communications (INFOCOM)*, pp. 226–234, 2015.
- [20] Sepideh Nazemi, Kin K. Leung and Ananthram Swami, "A Distributed, Energy-Efficient and QoI-Aware Framework for In-Network Processing", *IEEE International Conference on personal indoor and mobile radio communications (PIMRC)*, 2014.
- [21] Abhishek B. Sharma, Leana Golubchik, Ramesh Govindan, Michael J. Neely, "Dynamic data compression in multi-hop wireless networks", *SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1, pp. 145–156, 2009.
- [22] Marc Barcelo, Jaime Llorca, Antonia M Tulino, and Narayan Raman, "The cloud service distribution problem in distributed cloud networks", *IEEE Conference on Communications (ICC)*, pp. 344–350, 2015.
- [23] Hao Feng, Jaime Llorca, Antonia M Tulino, Danny Raz and Andreas F Molisch, "Approximation algorithms for the NFV service distribution problem", *IEEE Conference on Computer Communications (INFOCOM)*, pp. 1–9, 2017.
- [24] Ajmal Sawand, Soufiene Djahel, Zonghua Zhang and Farid Nait-Abdesselam, "Toward energy-efficient and trustworthy eHealth monitoring system" *China Communications*, vol. 12, no. 1, pp. 46–65, IEEE, 2015.
- [25] Georgios Tychogiorgos, Athanasios Gkelias and Kin K. Leung, "A Non-Convex Distributed Optimization Framework and its Application to Wireless Ad-hoc Networks", *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4286–4296, 2013.
- [26] Mung Chiang, *Geometric programming for communication systems*, Now Publishers Inc, 2005.
- [27] Stephen Boyd, Seung-Jean Kim, Lieven Vandenbergh and Arash Hassibi, "A tutorial on geometric programming", *Optimization and Engineering*, vol. 8, no. 1, pp. 67–127, Springer, 2007.
- [28] Dorit S Hochbaum, "Complexity and algorithms for nonlinear optimization problems", *Springer journal on Annals of Operations Research*, vol. 153, no. 1, pp. 257–296, 2007.
- [29] Dimitri P Bertsekas, *Nonlinear programming*, Athena scientific, 1999.
- [30] Stephen Boyd and Lieven Vandenbergh, *Convex optimization*, Cambridge University Press, 2004.
- [31] Sepideh Nazemi, Kin K. Leung and Ananthram Swami, "Optimization framework with reduced complexity for sensor networks with in-network processing", *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2016.
- [32] Sepideh Nazemi, Kin K. Leung and Ananthram Swami, "QoI-aware tradeoff between communication and computation in wireless ad-hoc networks", *IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–6, 2016.

APPENDIX A

OPTIMISATION PROBLEM WHEN INTERMEDIATE NODES

GENERATE DATA.

Forming the Lagrangian function of the problem (24) gives the following:

$$\begin{aligned}
 L(\delta, \lambda) = & \sum_{k \in K} y_k \left(f_k(\delta_k) + \sum_{i=0}^{h(k)-1} f_{k,i}(\delta_{k,i}) \prod_{m=i+1}^{h(k)} \delta_{k,m} \right) \\
 & + \sum_{i \notin K} \overset{\circ}{y}_i \left(f_i(\delta_i) + \sum_{z=0}^{h(i)-1} f_{i,z}(\delta_{i,z}) \prod_{m=z+1}^{h(i)} \delta_{i,m} \right) \\
 & - \lambda \left(\sum_{k \in K} y_k \left(\prod_{i=0}^{h(k)} \delta_{k,i} \right) + \sum_{i \notin K} \overset{\circ}{y}_i \left(\prod_{m=0}^{h(i)} \delta_{i,m} \right) + \gamma \right). \quad (34)
 \end{aligned}$$

Notice that the first two sums in (34) are similar in form, and can be replaced by a single summation over all nodes.

Comparing (34) with (11), we can immediately obtain (16). Equations (15) and hence (17) hold since the second summation in (34) does not involve the leaf node parameters. It is also clear that (18) will hold for the intermediate nodes, with y_j being interpreted as the total volume of incoming data: from the children and data that is sensed (generated) at that node. Hence (19) also holds.

APPENDIX B

PROOF OF LEMMA

Proof. Notice that equations (11)-(19) continue to hold for this special case of perfect c-ary tree with identical node parameters. We can write (19) in terms of previously defined parameters as

$$f'_j(\delta_j) = - \sum_{i=0}^{h(j)-1} f_{j,i}(\delta_{j,i}) \Delta_{j,m} + \Delta_j.$$

Parameters $\Delta_{j,m}$ and Δ_j only depend upon the δ 's of the ancestors of node j . Since the nodes are identical, the $f_{j,i}$'s are also identical. All intermediate nodes at hop distance 1 from the root node will then have the same $f'(\delta)$ value, and hence their optimal rate parameters will also be the same. We can repeat this argument level by level in the tree and conclude that nodes at a given level will have the same $f'(\delta)$ value and, under the assumption that $f'(\delta)$ is invertible, they will also have the same δ value. \square