Optimal Energy Tradeoff among Communication, Computation and Caching with QoI-Guarantee

Faheem Zafari[1,*], Jian Li[2,*], Kin K. Leung[3], Don Towsley[4] and Ananthram Swami[5]
[1,3]Imperial College London, [2,4]University of Massachusetts Amherst, [5]U.S. Army Research Laboratory
[1,3]{faheem16, kin.leung}@imperial.ac.uk, [2,4]{jianli, towsley}@cs.umass.edu, [5]ananthram.swami.civ@mail.mil
[*]Co-primary authors

***Abstract*—Energy efficiency is a fundamental requirement of modern data communication systems, and its importance is reflected in much recent work on performance analysis of system energy consumption. However, most works have only focused on communication and computation costs, but do not account for caching costs. Given the increasing interest in cache networks, this is a serious limitation. In this paper, we consider the energy consumption trade-off between communication, computation, and caching (C3) under a Quality of Information (QoI) guarantee in a communication network. To attain this goal, we formulate an optimization problem to capture the C3 costs, which turns out to be a non-convex Mixed Integer Non-Linear Programming (MINLP) Problem. We then propose a variant of spatial branch and bound algorithm (V-SBB), that can achieve ϵ-global optimal solution to the original MINLP. We show numerically that V-SBB is more stable and robust than other candidate MINLP solvers under different network scenarios. More importantly, we observe that the energy efficiency under our C3 optimization framework improves by as much as 88% compared to any C2 optimization between communication and computation or caching.**

I. Introduction

The rapid growth of smart environments, and advent of Internet of Things (IoT) have led to the generation of large amounts of data. However, it is a daunting task to transmit enormous amounts of data through traditional networks due to limited bandwidth and energy [1]. These data need to be efficiently compressed, transmitted, and cached to satisfy the Quality of Information (QoI) required by end users. In fact, many wireless components operate on limited battery power supply and are often deployed in remote or inaccessible areas, which necessitates the need for designs that can enhance the energy efficiency of the system with a QoI guarantee.

A particular example of modern systems that require high energy efficiency is the wireless sensor network (WSN). Consider a WSN with various types of sensors, which can generate enormous amount of data to serve end users. On the one hand, data compression has been adopted to reduce transmission (communication) cost at the expense of computation cost. On the other hand, caches can be used as a means of reducing transmission costs and access latency, thus enhancing QoI but with the expense of the added caching cost. Hence, there exists a tradeoff in energy consumption due to data communication, computation and caching. This raises the question: what is the right balance between compression and caching so as to minimize the total energy consumption of the network?

In this paper, we formulate an optimization problem that characterizes the tradeoff among communication, computation, and caching energy cost with QoI guarantee, and then develop an efficient algorithm to solve the optimization problem. Each node has the ability to compress and cache the data with some finite storage capacity. We focus on a tree-structured sensor network where each leaf node generates data, and compresses and transmits the data to the sink node in the network, which serves the requests for these data from devices outside this network. Examples of such a setting are military sites, wireless sensors in factories and hard-to-access sites, or societal networks, where a large number of devices gather data, and desire to transmit the local information to any device outside this network that requires this information. The objective of our work is to develop an efficient algorithm to minimize the total energy cost by incorporating data communication, computation and caching energy costs with a desired QoI constraint into our model, so that an optimal data compression rate at each node, and optimal caching locations in the network can be determined.

The main contributions of this paper are:

- This is the first attempt, to the best of our knowledge, to consider energy tradeoff among communication, computation and caching.
- We propose a variant of spatial branch-and-bound algorithm that provides ϵ-global[1] optimality guarantee that is more robust and stable to large variations in settings.
- Through extensive evaluations, we show that jointly optimizing communications, computations and caching can increase energy efficiency by as high as 88% when compared to optimizing only communications and either computation or caching.

This work was supported 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 copy-right notation hereon. Faheem Zafari also acknowledges the financial support by EPSRC Centre for Doctoral Training in High Performance Embedded and Distributed Systems (HiPEDS, Grant Reference EP/L016796/1), and Department of Electrical and Electronics Engineering, Imperial College London.

[1] ϵ-global optimality means that the obtained solution is within ϵ tolerance of the global optimal solution.

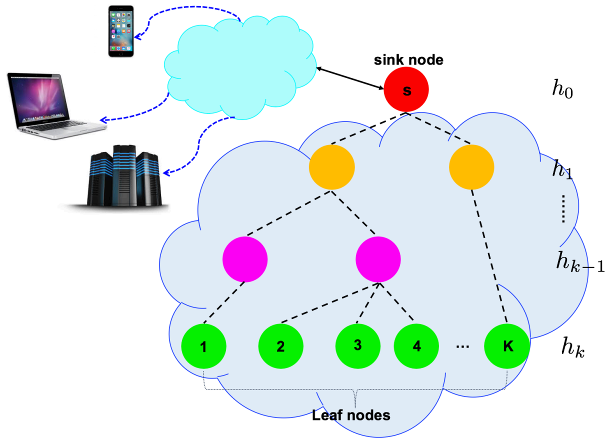

Fig. 1. Tree-Structured Network Model.

The paper is organized as follows. We describe our system model in Section II and formulate the problem of energy-efficient data compression, communication and caching with QoI constraint in Section III. We propose a variant of the Spatial Branch-and-Bound (V-SBB) algorithm in Section IV and evaluate the performance of our optimization framework and the proposed V-SBB algorithm through extensive numerical studies in Section V. Section VI describes the related work. We conclude the paper in Section VII.

II. Analytical Model

We represent the network as a directed graph $G = (V, E)$. Specifically, we consider a tree, with $N = |V|$ nodes, as shown in Figure 1. Node $v \in V$ is capable of storing S_v amount of data. Let $\mathcal{K} \subseteq V$ with $K = |\mathcal{K}|$ be the set of leaf nodes, with $\mathcal{K} = \{1, 2, \cdots, K\}$. Time is partitioned in periods of equal length $T > 0$ and data generated in each period are independent, i.e., the transmitted and cached data in one time period may be totally replaced with new data for the next time period[2]. W.l.o.g., we consider one particular period in the remainder of the paper. We assume that only leaf nodes $k \in \mathcal{K}$ can generate data, and all other nodes in the tree receive and compress data from their children nodes, and either cache or transmit the compressed data to their parent nodes during time T. Relaxation of the preceding assumptions is discussed in Section III-B.

Let y_k be the amount of data generated by leaf node $k \in \mathcal{K}$. The data generated at the leaf nodes are transmitted up the tree to the sink node s, which serves the requests for the data generated in the network. Let $h(k)$ be the depth of node k in the tree. W.l.o.g., we assume that the sink node is located at level $h(s) = 0$. We represent the unique path from node k to the sink node by $\mathcal{H}^k$ of length $h(k)$ the sequence $\{h_0^k, h_1^k, \cdots, h_{h(k)}^k\}$ of nodes $h_j^k \in V$ such that $(h_j^k, h_{j+1}^k) \in E$, where $h_0^k \triangleq s$ (i.e., the sink node) and $h_{h(k)}^k \triangleq k$ (i.e., the node itself).

We denote the per-bit reception, transmission and compression cost of node $v \in V$ as $\varepsilon_{vR}, \varepsilon_{vT}$, and ε_{vC}, respectively. Each node h_i^k along the path $\mathcal{H}^k$ can compress the data generated by leaf node k with a *data reduction rate* $\delta_{k,i}$ (ratio of the outgoing data from a node to the incoming data), where $0 < \delta_{k,i} \leq 1, \forall i, k$. The reduction rate characterizes the degree to which a node can compress the received data, which plays an important role for determining the QoI.

The higher the value of $\delta_{k,i}$, the lower the compression will be, and vice versa. The higher the degree of data compression, the larger will be the amount of energy consumed by compression. Similarly, caching the data closer to the sink node may reduce the transmission cost for serving the request, however, each node only has finite storage capacity. We study the tradeoff among the energy consumed at each node for transmitting, compressing and caching the data.

Denote the total energy consumption at node v as E_v, which consists of the reception cost E_{vR}, transmission cost E_{vT}, computation cost E_{vC} and storage (caching) cost E_{vS}[3]; it takes the form

$$E_v = E_{vR} + E_{vT} + E_{vC} + E_{vS}, \tag{1}$$

where

$$\begin{aligned} E_{vR} &= y_v \varepsilon_{vR}, & E_{vT} &= y_v \varepsilon_{vT} \delta_v, \\ E_{vC} &= y_v \varepsilon_{vC} l_v(\delta_v), & E_{vS} &= w_{ca} y_v T. \end{aligned} \tag{2}$$

Here, $l_v(\delta_v)$ captures the computation energy. As computation energy increases with the degree of compression, we assume that $l_v(\delta_v)$ is a continuous, decreasing and differentiable function of the reduction rate. One candidate function is $l_v(\delta_v) = 1/\delta_v - 1$ [1], [2]. Moreover, we consider an energy-proportional model [3] for caching, i.e., $E_{vS} = w_{ca} y_v T$ if the received data y_v is cached for a duration of T where w_{ca} represents the power efficiency of caching, which strongly depends on the storage hardware technology. W.l.o.g., w_{ca} is assumed to be identical for all the nodes. For simplicity, let

$$f(\delta_v) = \varepsilon_{vR} + \varepsilon_{vC} l_v(\delta_v) + \varepsilon_{vT} \delta_v, \tag{3}$$

which is the sum of per-bit reception, compression and transmission (RCT) cost at node v per unit time.

During time period T, we assume that there are R_k requests at the sink node s for data y_k generated by leaf node k[4]. We set the boolean variable $b_{k,i}$ to 1 if the data from node k is stored along the path $\mathcal{H}^k$ at node h_i^k, otherwise it equals 0. At most, only a single copy of data from node k is stored along the path $\mathcal{H}^k$. For ease of notation, we denote $b_{k,h(k)}$ by b_k, $f_{k,h(k)} \triangleq f_k$ and $\delta_{k,h(k)} \triangleq \delta_k$. Let C_v denote the set of leaf nodes $k \in \mathcal{K}$ that are descendants of node v. We also assume that the energy cost for searching for data at different nodes in the network is negligible [1], [4].

[2] In some literature, this is called *windowed grouped data aggregation* where data generated in a finite time period must be compressed.

[3] Provided that data is cached.

[4] As motivated in Section I, a large number of agents may desire the same information, hence there are multiple requests for the same data.

III. Energy Optimization

In this section, we first define the cost function in our model and then formulate the optimization problem. Data produced by every leaf node is received, transmitted, and possibly compressed by all nodes in the path from the leaf node to the root node, consuming energy

$$E_k^{\mathrm{C}} = \sum_{i=0}^{h(k)} y_k f(\delta_{k,i}) \prod_{m=i+1}^{h(k)} \delta_{k,m}, \tag{4}$$

where $\prod_{m=i}^{j} \delta_{k,m} := 1$ if $i \geq j$. Equation (4) captures one-time[5] energy cost of receiving, compressing and transmitting data y_k from leaf node (level $h(k)$) to the sink node (level 0). The amount of data received by any node at level i from leaf node k is $y_k \prod_{m=i+1}^{h(k)} \delta_{k,m}$ due to the compression from level $h(k)$ to $i+1$. The term $f(\delta_{k,i})$ captures the reception, compression, and transmission (RCT) energy cost for node at level i along the path from leaf node k to the sink node.

Let E_k^{R} be the total energy consumed in responding to the subsequent $(R_k - 1)$ requests. We have

$$E_k^{\mathrm{R}} = \sum_{i=0}^{h(k)} y_k (R_k - 1) \Bigg\{ f(\delta_{k,i}) \prod_{m=i+1}^{h(k)} \delta_{k,m} \Big(1 - \sum_{j=0}^{i} b_{k,j}\Big) + \Big(\prod_{m=i}^{h(k)} \delta_{k,m}\Big) b_{k,i} \left(\frac{w_{ca} T}{R_k - 1} + \varepsilon_{kT}\right) \Bigg\}. \tag{5}$$

Note that the remaining $(R_k - 1)$ requests are either served by the leaf node or a cached copy of data y_k at level i for $i = 1, \cdots, h(k)$. W.l.o.g., we consider node $v_{k,i}$ at level i. If data y_k is not cached from $v_{k,i}$ up to the sink node (level 0), i.e., $b_{k,j} = 0$ for $j = 0, \cdots, i$, cost is incurred due to receiving, transmitting and compressing the data $(R_k - 1)$ times, which is captured by the first term in Equation (5), the second term is 0. Otherwise, the $(R_k - 1)$ requests are served by the cached copy at $v_{k,i}$, the corresponding caching and transmission cost serving from $v_{k,i}$ are captured by the second term in Equation (5), and the corresponding RCT cost from $v_{k,i-1}$ upto to sink node is captured by the first term. Note that the first time cost of receiving, compressing and transmitting the data from leaf node to $v_{k,i}$ is already captured by Equation (4). A simple but illustrative example to explain the above equations is provided in [5]. The total energy consumed in the network is E^{total},

$$E^{\text{total}}(\boldsymbol{\delta}, \boldsymbol{b}) \triangleq \sum_{k \in \mathcal{K}} \Big(E_k^{\mathrm{C}} + E_k^{\mathrm{R}}\Big), \tag{6}$$

where $\boldsymbol{\delta} = \{\delta_{k,i}, \forall k \in \mathcal{K}, i = 0, \cdots, h(k)\}$ and $\boldsymbol{b} = \{b_{k,i}, \forall k \in \mathcal{K}, i = 0, \cdots, h(k)\}$. Our objective is to minimize the total energy consumption of the network with a QoI constraint for end users by choosing the compression ratio vector $\boldsymbol{\delta}$ and caching decision vector $\boldsymbol{b}$ in the network G. Therefore, the optimization problem is,

$$\begin{aligned}
\min_{\boldsymbol{\delta}, \boldsymbol{b}} \quad & E^{\text{total}}(\boldsymbol{\delta}, \boldsymbol{b}) \\
\text{s.t.} \quad & \sum_{k \in \mathcal{K}} y_k \prod_{i=0}^{h(k)} \delta_{k,i} \geq \gamma, \\
& b_{k,i} \in \{0, 1\}, \forall k \in \mathcal{K}, i = 0, \cdots, h(k), \\
& \sum_{k \in C_v} b_{k,h(v)} y_k \prod_{j=h(k)}^{h(v)} \delta_{k,j} \leq S_v, \forall\, v \in V, \\
& \sum_{i=0}^{h(k)} b_{k,i} \leq 1, \forall k \in \mathcal{K},
\end{aligned} \tag{7}$$

where $h(v)$ is the depth of node v in the tree.

The first constraint is the QoI constraint, i.e., the total data available at the sink node [1]. The second constraint indicates that our decision (caching) variable $b_{k,i}$ is binary. The third constraint is on total amount of data that can be cached at each node. The fourth constraint is that at most one copy of the generated data should be cached on the path between the leaf node and the sink node. The optimization problem in (7) is a non-convex MINLP problem with M continuous variables, the $\delta_{k,i}$'s and M binary variables, the $b_{k,i}$'s where, $M = \sum_{k \in \mathcal{K}} h(k)$.

A. Properties

Theorem 1. *The optimization problem* (7) *is NP-hard.*

Proof. The optimization problem (7) can be reduced to a general non-convex MINLP problem. Due to space limitations, the general form of a non-convex MINLP and the reduction steps are presented in [5]. Since non-convex MINLP is NP-hard [6], the optimization problem described in (7) is NP-hard. □

Remark 1. *The objective function E^{total} defined in* (7) *is monotonically increasing in the number of requests R_k for all $k \in \mathcal{K}$ provided that $\boldsymbol{\delta}$ and $\boldsymbol{b}$ are fixed.*

Notice that (4) is independent of R_k and (5) is linear in R_k, and its multipliers are positive. Hence, for any fixed $\boldsymbol{b}$ and $\boldsymbol{\delta}$, (6) increases monotonically with R_k.

Remark 2. *Given a fixed network scenario, if we increase the number of requests R_k for the data generated by leaf node k, then these data will be cached closer to the sink node or at the sink node, if there exists enough cache capacity, to reduce the overall energy consumption.*

For fixed $\boldsymbol{\delta}$, observe from (5) that energy consumption decreases if the cache is moved closer to the root as the nodes deep in the tree do not need to retransmit.

B. Relaxation of Assumptions

In our model, we make several assumptions for the sake of simplicity. While we assume that the network is structured as a tree, this assumption can be easily relaxed as long as there

[5] During every time period T, data is always pushed towards the sink upon the first request.

exists a simple fixed path from each leaf node to the sink node. The tree structure represents a simple topology that captures the key parameters in the optimization formulation without the complexity introduced by a general network topology. Furthermore, for simplicity, we assume that all parameters across the nodes are identical, which is not necessary as seen from the cost function. We also assume that only leaf nodes generate data. However, our model can be extended to allow intermediate nodes to generate data at the cost of added complexity.

IV. Variant of Spatial Branch-and-Bound Algorithm

We present a variant of the Spatial Branch-and-Bound algorithm (V-SBB). Instead of solving the MINLP problem in (7) directly, we use V-SBB to solve a *standard form*[6] of the original MINLP which is equivalent to the problem in (7).

A. Variant of Spatial Branch-and-Bound Algorithm

In contrast to the conventional SBB, our newly proposed V-SBB eliminates the bound-tightening steps. This takes care of two issues in SBB: (i) bound tightening step does not always guarantee faster convergence; (ii) removal significantly reduces the computational complexity of the algorithm. Algorithm 1 gives an overview of V-SBB.

We briefly describe the key steps due to space limitations. A detailed explanation of each step is given in [5].

Step 2: We use the *least lower bound rule*[7] to choose a subregion $\mathcal{R}$ from $\mathcal{L}$ among all feasible subregions. This lower bound is obtained by solving a convex relaxation of the reformulated problem. McCormick linear over-estimators and underestimators [8] are used to obtain the convex relaxation for bilinear terms (bt) and linear fractional terms (lft). Denote the optimal solution of this subregion as $\phi^{\mathcal{R},l}$. Note that if the convex relaxation is infeasible or the obtained lower bound is greater than the current upper bound ϕ^u, we move to *Step* 5, otherwise we move to *Step* 3.

Step 3: We compute the upper bound $\phi^{\mathcal{R},u}$ for the subregion $\mathcal{R}$ through local MINLP solver such as *Bonmin* [9]. If this upper bound cannot be obtained or is greater than ϕ^u, we move to *Step* 4. Otherwise, we set it as the current best solution ϕ^u, and delete all other subregions that have higher lower bounds than this region's upper bound. If the difference between the upper and lower bounds for this subregion is within ϵ-tolerance, we delete this subregion by moving to *Step* 5, otherwise move to *Step* 4.

Step 4: known as the *branching* step, is used to select a variable and its corresponding value at which the region is further divided. Here, we use the variable and value selection rule specified in [7], under which the variable that causes the maximal reduction in the feasibility gap between the solution of *Step* 2 and the exact problem, is branched on. Then we partition $\mathcal{R}$ into $\mathcal{R}_{\text{right}}$ and $\mathcal{R}_{\text{left}}$, and add them into $\mathcal{L}$ as well as delete $\mathcal{R}$.

B. Convergence of V-SBB

Theorem 2. *Our V-SBB described in Algorithm 1 converges to an ϵ-global optimal solution of its standard problem given in [5].*

Though we made critical modifications to obtain our V-SBB algorithm, the proof of convergence follows an argument similar to that of *Branch-and-Select* given in [10]. We present the poof in [5] for completeness.

Algorithm 1 Variant of Spatial Branch-and-Bound (V-SBB)

```
Step 1: Initialize $\phi^u := \infty$ and $\mathcal{L}$ to a single domain
Step 2: Choose a subregion $\mathcal{R} \in \mathcal{L}$ using least lower bound rule
if $\mathcal{L} = \emptyset$ then Go to Step 6
if for chosen region $\mathcal{R}$, $\phi^{\mathcal{R},l}$ is infeasible or $\phi^{\mathcal{R},l} \geq \phi^u - \epsilon$ then Go to Step 5
Step 3: Obtain upper bound $\phi^{\mathcal{R},u}$
if upper bound cannot be obtained or if $\phi^{\mathcal{R},u} > \phi^u$ then Go to Step 4
else $\phi^u := \phi^{\mathcal{R},u}$ and, from the list $\mathcal{L}$, delete all subregions $\mathcal{S} \in \mathcal{L}$ such that $\phi^{\mathcal{S},l} \geq \phi^u - \epsilon$
if $\phi^{\mathcal{R},u} - \phi^{\mathcal{R},l} \leq \epsilon$ then Go to Step 5
Step 4: Partition $\mathcal{R}$ into new subregions $\mathcal{R}_{\text{right}}$ and $\mathcal{R}_{\text{left}}$
Step 5: Delete $\mathcal{R}$ from $\mathcal{L}$ and go to Step 2
Step 6: Terminate Search
if $\phi^u = \infty$ then Problem is infeasible
else $\phi^u$ is $\epsilon$-global optimal
```

V. Evaluation

We evaluate the performance of our V-SBB algorithm as well as the energy efficiency of our communication, compression and caching (C3) joint optimization framework through a series of experiments on several network topologies as shown in Figure 2. Our key objective is to gain preliminary insights into our algorithm when compared with a few other well-known techniques. The highlights of the evaluation results are:

(1) Our V-SBB algorithm can obtain an ϵ-global optimal solution within a reasonable time in most situations. Also it is robust and stable to various parameters in different network scenarios.

(2) When Bonmin [9] can achieve a solution, it is faster. However, the solution obtained through Bonmin is not always comparable to that of V-SBB. We observe that when higher compression is done (i.e., smaller value of γ), V-SBB always outperforms Bonmin. More importantly, we find that Bonmin has poor performance in stability and robustness, i.e., it cannot even produce feasible solutions in some cases although they exist. NOMAD [11] and GA [12] often produce objective-function values much larger than V-SBB.

(3) Our C3 joint optimization framework improves energy

[6] We reformulate the MINLP (7) into a standard form needed by V-SBB using an approach called *Symbolic Reformulation* [7]. We omit the details here due to space constraint and refer readers to [5] for a detailed discussion.

[7] Select a subregion $\mathcal{R} \in \mathcal{L}$, whose convex relaxation provides the lowest objective function value.

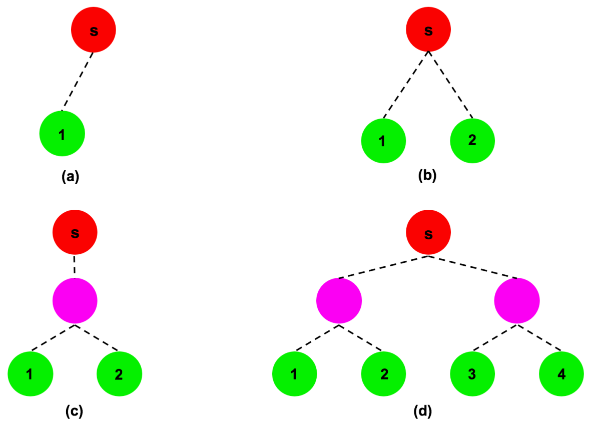

Fig. 2. Candidate network topologies used in the experiments

efficiency by as much as 88% compared to the C2 optimization over communication and computation, or communication and caching.

A. Methodology

Performance metrics: Our primary metrics for comparisons are:

(1) The best solution to the objective function: Since obtaining the global optimum for the NP-hard problem is daunting, we are primarily interested in ϵ-global optimum;

(2) Convergence Time, which is the time an algorithm needs to obtain the best solution;

(3) Stability and Robustness, which is characterized by the frequency or ability of the algorithm to provide feasible solutions, provided that they are known to exist;

(4) Energy efficiency in joint optimization. We compare the energy cost of our joint optimization framework for communication, computation and caching (C3) with that of the optimization of any of the two types of resources (denoted by C2) under the same situation. The energy efficiency $\mathcal{E}$ defined as:

$$\mathcal{E} = \frac{E^{\text{total}*}(\text{C2}) - E^{\text{total}*}(\text{C3})}{E^{\text{total}*}(\text{C2})} \times 100\%, \tag{8}$$

where $E^{\text{total}*}(\text{C3})$ and $E^{\text{total}*}(\text{C2})$ are the optimal energy costs under the C3 optimization framework in (7) and the C2 optimization, respectively. $\mathcal{E}$ reflects the increase of energy efficiency for the C3 over the C2 optimization.

Setup: We implement V-SBB in Matlab on a Core i7 3.40 GHz CPU with 16 GB RAM. The candidate MINLP solvers in this work include Bonmin, NOMAD and GA, which are implemented with Opti-Toolbox [13]. The reformulations needed are executed by a Java based module and we derive the bounds on the auxiliary variables. V-SBB terminates when ϵ-optimality is obtained or a computation timer of 200 seconds expires. We take $\epsilon = 0.001$ in our study. If the timer expires, the last feasible solution is taken as the best solution. Our simulation parameters are provided in Table I, which are the typical values used in the literature [1], [14].

TABLE I
PARAMETERS USED IN SIMULATIONS

Parameter	Value
y_k: Amount of data produced by each leaf node k	1000
R_k: Request rate for data from node k	100
w_{ca}: Caching power efficiency	1.88×10^{-6}
T: Time length that data are cached	10s
ε_{vR}: Per-bit reception cost of node v	50×10^{-9}
ε_{vT}: Per-bit transmission cost of node v	200×10^{-9}
ε_{vC}: Per-bit compression cost of node v	80×10^{-9}
γ: QoI threshold	$[1, \sum_{k\in\mathcal{K}} y_k]$
S_v: Storage capacity of node v	1000

B. The Best Solution to the Objective Function

We compare the performance of V-SBB with three other candidate solvers for the networks in Figure 2. The results for two nodes and seven nodes are presented in Table II. We also relax the integer constraint in (7) to be continuous, i.e., $b_{k,i} \in [0,1]$. Then (7) becomes a geometric programming problem, which can be solved by IPOPT [15]. We call it "Relaxed" and the corresponding results are presented in the last row of Table II. We observe that V-SBB achieves the lowest value comparable to Bonmin for larger values of γ, and significantly outperforms Bonmin for smaller values of γ, which we discuss in detail later. However, Bonmin cannot generate a feasible solution even if it exists for some cases. This is because Bonmin is built on the Branch-and-Cut method, which sometimes cuts regions where a lower value exists. NOMAD and GA in general yield a higher objective-function value than V-SBB does. Similar trends are observed for three and four node networks, details deferred to [5] due to space limitations.

Figure 3 shows that the optimal energy cost is monotonically increasing with the number of requests for a two node and seven node network. The results are obtained using our C3 framework for $\gamma = 0.25\sum_{k\in\mathcal{K}} y_k$ and $\gamma = 0.75\sum_{k\in\mathcal{K}} y_k$, respectively. For the network parameters under consideration, we note that there is a turning point on the curves, and the total energy cost increases much faster with the number of requests before the turning point than that after it. This is because the data has already been cached at the root node at this point and there is no need to retrieve data from other nodes in the network, which reduces transmission costs. This is the benefit that caching brings, and we will further discuss the advantage of C3 optimization over the C2 later in Section V-E.

C. Convergence Time

The amount of time that an algorithm requires to obtain its best solution as discussed in Section V-B are shown in Table II for the two node and seven node networks, respectively. It can be see that Bonmin is the fastest method since it uses the branch-and-cut approach which cuts certain domains to accelerate the branching process. As discussed earlier, the Bonmin algorithm is fast at the expense of algorithm stability, i.e., sometimes it cannot find a solution although it exists. V-SBB takes longer to obtain a better solution, because our reformulation introduces auxiliary variables and additional linear constraints.

TABLE II

The Best Solution to the Objective Function (Obj.) and Convergence time for seven nodes network

Solver	$\gamma = 1$		$\gamma = 1000$		$\gamma = 2000$		$\gamma = 3000$		$\gamma = 4000$	
	Obj.	**Time** (s)	**Obj.**	**Time** (s)	**Obj.**	**Time** (s)	**Obj.**	**Time** (s)	**Obj.**	**Time** (s)
Bonmin	0.0002	0.214	0.039	0.164	0.078	0.593	0.117	0.167	0.156	0.212
NOMAD	0.004	433.988	0.121	381.293	0.108	203.696	0.158	61.093	0.181	26.031
GA	0.043	44.538	0.096	30.605	0.164	44.970	0.226	17.307	0.303	28.820
V-SBB	0.0001	1871.403	0.039	25.101	0.078	30.425	0.117	23.706	0.156	19.125
Relaxed	0.0002	0.201	0.039	0.111	0.078	0.095	0.117	0.102	0.156	0.105

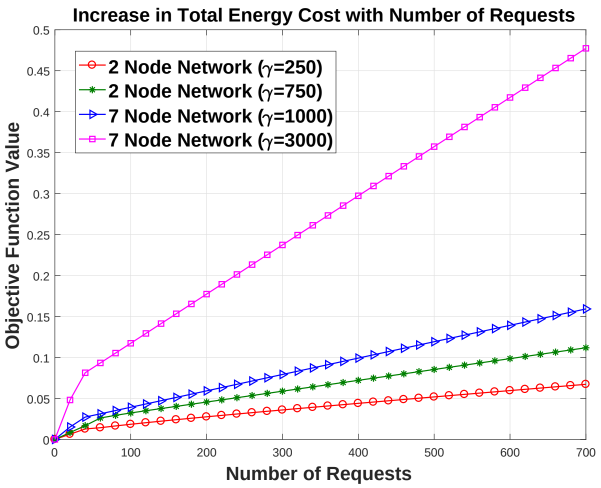

Fig. 3. Total Energy Costs vs. Number of Requests.

TABLE III

Infeasibility of Bonmin for networks in Figure 2

Networks	(a)	(b)	(c)	(d)
$\#$ **of test values**	1000	2000	2000	4000
$\#$ **of infeasible solutions**	0	0	1	216
Infeasibility (%)	0	0	0.05	5.4

D. Stability and Robustness

From the analysis in Sections V-B and V-C, we know that Bonmin is faster but unstable in some situations. We further characterize the stability of Bonmin with respect to the threshold value of QoI γ as follows. Specifically, we fix all other parameters in Table I, and vary only the maximal possible value of γ in different networks. The results are shown in Table III. We see that the number of instances where the Bonmin method fails to produce a feasible solution increases as the network size increases. This is mainly due to the cutting phase in the Bonmin method, which cuts the feasible regions that need to be branched.

Although Bonmin can provide a feasible solution for smaller values of γ in less time, we observe that the value of the solution is larger than that of V-SBB. We compare the performance of V-SBB and Bonmin for smaller values of γ in Table IV. We see that V-SBB outperforms Bonmin by as much as 52.45% when searching for an ϵ-global optimum, though it requires more time. The timer is set to 7200s for results shown in Table IV.

E. Energy Efficiency

We compare the total energy costs under joint C3 optimization with those under C2 optimization. We consider two cases for the C2 optimization: (i) C2o (Communication and Computation), where we set $S_v = 0$ for each node to avoid any data caching; (ii) C2a (Communication and Caching), where we set $\gamma = \sum_{k\in\mathcal{K}} y_k$, which is equivalent to $\delta_v = 1, \forall v \in V$, i.e., no compression. Comparison between C3, C2o and C2a is shown in Figure 4.

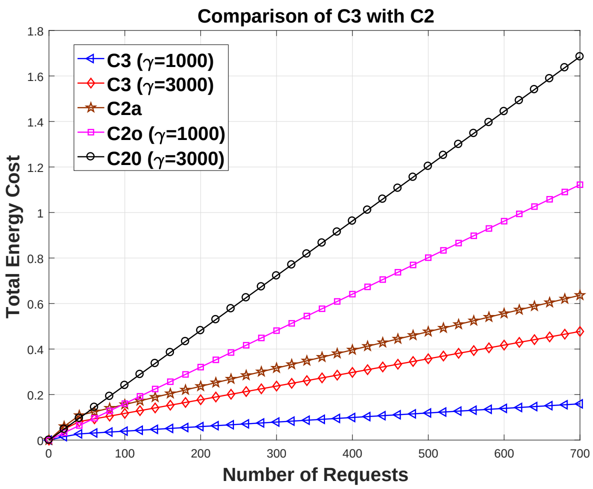

Fig. 4. Comparison of C3 and C2 optimization for the seven node network in Figure 2.

First, we observe that as the number of requests increases, the total energy cost increases. Second, the energy cost for the C3 joint optimization is lower than that for C2o optimization for the same parameter setting. This captures the tradeoff between caching, communication and computation. In other words, although C3 incurs caching costs, it may significantly reduce the communication and computation, which in turn brings down total energy cost. Similarly, C3 optimization outperforms C2a although C3 incurs caching cost. Using Equation (8), energy efficiency improves by as much as 88% for the C3 framework when compared with the C2 formulation. These trends are observed in other candidate network topologies and readers are referred to [5] for details due to space limitations.

Remark 3. *From our analysis, it is clear that the larger the ratio between ε_{vT} and ε_{vR}, ε_{vC}, the larger will be the improvement provided by our C3 formulation.*

VI. Related Work

To the best of our knowledge, there is no prior work that jointly considers communication, computation and caching costs in distributed networks with a QoI guarantee for end users.

Data Compression: Compression is a key operation in modern communication networks and has been supported by many

TABLE IV
COMPARISON BETWEEN V-SBB AND BONMIN FOR SMALLER VALUES OF γ IN SEVEN NODE NETWORK

Solver	γ =1		γ=3		γ =5		γ =8		γ =50	
	Obj.	**Time (s)**	**Obj.**	**Time**	**Obj.**	**Time**	**Obj.**	**Time**	**Obj.**	**Time**
Bonmin	0.0002	0.214	0.0003	0.211	0.0003	0.224	0.0005	0.23	0.0021	0.364
V-SBB	0.00011	1871.403	0.00015	2330	0.00019	1243.77	0.00047	1350.016	0.0020	3325.302
Improvement (%)	52.45		49.43		50.30		7.59		4.62	

data-parallel programming models [16]. For WSNs, data compression is usually performed over a hierarchical topology to improve communication energy efficiency [17], whereas we focus on energy tradeoff between communication, computation and caching.

Data Caching: Caching plays a significant role in many systems with hierarchical topologies, e.g., WSNs, microprocessors, CDNs etc. There is a rich literature on the performance of caching in terms of designing different caching algorithms, e.g., [4], [18], and we do not attempt to provide an overview here. Utility maximization approach has also been studied for cache management [19]–[22]. However, none of these work considered the costs of caching, which may be significant in some systems [3]. The recent paper by Li et al. [23] is closest to the problem we tackle here. The differences between our work and [23] are mainly from two perspectives. First, the mathematical formulations are quite different, we consider energy tradeoffs between C3 while [23] focused on C2. Second, we provide a ϵ-optimal solution to a MINLP problem while [23] aimed at developing approximation algorithms.

Energy Costs: While optimizing energy costs in wireless sensor networks has been extensively studied [14], existing work primarily is concerned with routing [24], MAC protocols [14], and clustering [25]. With the growing deployment of smart sensors in modern systems [1], in-network data processing, such as data aggregation, has been widely used as a mean of reducing system energy cost by lowering the data volume for transmission.

VII. CONCLUSION

We have investigated energy efficiency tradeoffs among communication, computation and caching with QoI guarantee in communication networks. We first formulated an optimization problem that characterizes these energy costs and showed that the problem is non-convex MINLP i.e. NP-hard. We then proposed a variant of the spatial branch-and-bound (V-SBB) algorithm, which can solve the MINLP with ϵ-optimality guarantee. Finally, we showed numerically that the proposed V-SBB algorithm outperforms existing MINLP solvers, Bonmin, NOMAD and GA. We also observed that C3 optimization framework, leads to an energy saving of as much as 88% when compared with either of the C2 optimizations which have been widely studied. In future, we aim to optimize the convergence time for smaller values of γ and reduce the computational complexity of our approach so that it can be applied to larger networks.

REFERENCES

[1] S. Nazemi, K. K. Leung, and A. Swami, "QoI-aware Tradeoff Between Communication and Computation in Wireless Ad-hoc Networks," in *Proc. IEEE PIMRC*, 2016.

[2] S. Eswaran, J. Edwards, A. Misra, and T. F. L. Porta, "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, Sept 2012.

[3] N. Choi, K. Guan, D. C. Kilper, and G. Atkinson, "In-network Caching Effect on Optimal Energy Consumption in Content-Centric Networking," in *Proc. IEEE ICC*, 2012.

[4] S. Ioannidis and E. Yeh, "Adaptive Caching Networks with Optimality Guarantees," in *Proc. of ACM SIGMETRICS*, 2016.

[5] F. Zafari, J. Li, K. K. Leung, D. Towsley, and A. Swami, "Optimal Energy Tradeoff among Communication, Computation and Caching with QoI-Guarantee," *Arxiv preprint arXiv:1712.03565*, 2017.

[6] C. A. Floudas, *Deterministic Global Optimization: Theory, Methods and Applications*. Springer Science & Business Media, 2013, vol. 37.

[7] E. M. Smith, "On the Optimal Design of Continuous Processes," Ph.D. dissertation, Imperial College London (University of London), 1996.

[8] G. P. McCormick, "Computability of Global Solutions to Factorable Nonconvex Programs: Part IConvex Underestimating Problems," *Mathematical Programming*, vol. 10, no. 1, pp. 147–175, 1976.

[9] P. Bonami *et al.*, "An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs," *Disc. Opt.*, vol. 5, no. 2, pp. 186–204, 2008.

[10] L. Liberti, "Reformulation and Convex Relaxation Techniques for Global Optimization," Ph.D. dissertation, Imperial College London, 2004.

[11] S. Le Digabel, "Algorithm 909: NOMAD: Nonlinear Optimization with the MADS Algorithm," *ACM TOMS*, vol. 37, no. 4, p. 44, 2011.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[13] OPTI Toolbox, "A Free Matlab Toolbox for Optimization," https://www.inverseproblem.co.nz/OPTI/index.php/Main/HomePage, [Online; accessed 28-Jun-2017].

[14] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *System sciences*, 2000.

[15] A. Wächter and L. T. Biegler, "On the Implementation of an Interior-point Filter Line-search Algorithm for Large-scale Nonlinear Programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[16] O. Boykin, S. Ritchie, I. O'Connell, and J. Lin, "Summingbird: A Framework for Integrating Batch and Online Mapreduce Computations," *Proc. of VLDB*, 2014.

[17] R. Rajagopalan and P. K. Varshney, "Data Aggregation Techniques in Sensor Networks: A Survey," *IEEE Commun. Surveys Tuts.*, vol. 8, no. 4, pp. 48–63, 2006.

[18] J. Li, S. Shakkottai, J. C. S. Lui, and V. Subramanian, "Accurate Learning or Fast Mixing? Dynamic Adaptability of Caching Algorithms," *IEEE Journal on Selected Areas in Communications*, 2018.

[19] M. Dehghan, L. Massoulie, D. Towsley, D. Menasche, and Y. Tay, "A Utility Optimization Approach to Network Cache Design," in *Proc. of IEEE INFOCOM*, 2016.

[20] N. K. Panigrahy, J. Li, and D. Towsley, "Hit Rate vs. Hit Probability Based Cache Utility Maximization," in *Proc. of ACM MAMA*, 2017.

[21] N. K. Panigrahy, J. Li, F. Zafari, D. Towsley, and P. Yu, "Optimizing Timer-based Policies for General Cache Networks," *Arxiv preprint arXiv:1711.03941*, 2017.

[22] N. K. Panigrahy, J. Li, and D. Towsley, "Network Cache Design under Stationary Requests: Exact Analysis and Poisson Approximation," *Proc. of IEEE MASCOTS*, 2018.

[23] J. Li, F. Zafari, D. Towsley, K. K. Leung, and A. Swami, "Joint Data Compression and Caching: Approaching Optimality with Guarantees," in *Proc. of ACM/SPEC ICPE*, 2018.

[24] A. Manjeshwar and D. P. Agrawal, "TEEN: a Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks," in *IPDPS*, 2001.

[25] M. Ye, C. Li, G. Chen, and J. Wu, "EECS: an Energy Efficient Clustering Scheme in Wireless Sensor Networks," in *Proc. of IEEE IPCCC*, 2005.