Jointly Compressing and Caching Data in Wireless Sensor Networks

Nitish K. Panigrahy[1,*], Jian Li[2,*], Faheem Zafari[3], Don Towsley[4] and Paul Yu[5]
[1,2,4]University of Massachusetts Amherst, [3]Imperial College London, [5]U.S. Army Research Laboratory
[1,2,4]{nitish, jianli, towsley}@cs.umass.edu, [3]faheem16@imperial.ac.uk, [5]paul.l.yu.civ@mail.mil

***Abstract*—We propose a novel policy for data compression and caching in a wireless sensor network (WSN) that provably optimizes utility and cost jointly, providing a theoretical basis to understand the compression-caching tradeoff for data analytics in a WSN. Our optimization framework provides analytical answers to how much compression should be performed at each sensor, and where the data should be cached in the network. We propose a distributed algorithm to implement the optimal policy and adapt to the changes (e.g., cache size and request processes) in the network. We evaluate our approach through extensive simulations on WSNs.**

I. Introduction

Nowadays, a large amount of data is continuously generated by distributed sources such as smart-wearables, sensors, and Internet-of-Thing devices in many applications [1]. For example, services like Facebook, Twitter and Netflix continuously gather these data from agents for many analytical purposes, like finding popular contents among agents, and popular words in tweets. In this paper, we focus on such data analytics in a wireless sensor network (WSN).

Tree-structured model. A typical analytics infrastructure for processing such data streams in a WSN usually follow a tree structure (See Figure 1), which conceptually comprises a single centralized "sink" node connected to multiple end sensors through many routers by a WSN. The data is usually generated at end sensors, and then sent through routers to the sink node. Analysts make their queries to retrieve relevant data at the sink. Without loss of generality (w.l.o.g.), we assume that there is a path from each end sensor to the sink node, consisting of multiple routers between them. Note that in such a WSN, only end sensors generate data.

Data compression. Various data compression algorithms have been developed in a WSN, which are widely applied to medical imaging, cameras, and video-on-demand systems [2]. Furthermore, data summarization can be viewed as compression. By adopting data compression, data processing can be pushed towards the edge to assign tags/labels, pull meta data and answer queries. In this work we focus on optimizing data compression in a WSN. We consider a case where each router j can compress the incoming data i generated by sensor k with a reduction rate $\delta_{ij}^{(k)}$, where $0 < \delta_{ij}^{(k)} \leq 1$. A smaller $\delta_{ij}^{(k)}$ value means that data from sensor k is compressed to a larger extent at node j, resulting in lower data quality at a higher energy cost.

*Authors with equal contribution.

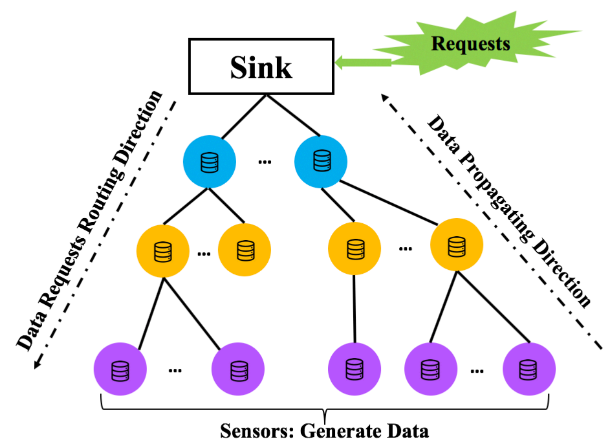

Fig. 1. An illustrative example for data generating, request and propagating in a wireless sensor network.

TTL-caches. Caching data in a WSN can be beneficial in the case when multiple requesters with different processing capabilities have different requirements from a single content. For example, from a received image, one requester may look for vehicle information while another may look for people. Also if a sensor dies, it may be useful for its data to be cached for later consumption. In this work each router/end sensor is associated with a cache to store a finite amount of incoming data for compression. A Time-to-Live (TTL) value T_{ij} is associated with every data i at router j. T_{ij} basically indicates the time period for which data i must be kept in the cache associated with router j. This will be described in detail in Section II.

Compression-Caching tradeoff. Data transfer from router to router happens over WSN links, which are generally scarce or expensive. To save bandwidth, incoming data at each router is compressed before sending it to the next router on the path, or to be cached in routers. Such data compression and caching along the path leads to a fundamental tradeoff between two key metrics: *utility* and *cost*. Here, utility corresponds to user satisfaction on their requests, which is a function of "hit probability" and "reduction rate", while cost is caused by routing the compressed data from the end sensor to sink node to satisfy analysts' requests. These will be described in details in Section III. In general, a smaller reduction rate incurs a lower cost but results in a lower utility. On the other hand, caching the compressed data at routers closer to the sink achieves higher utility but results at higher cost. However, the cache associated with each router can only store a finite amount of data. *A goal of this work is to devise optimal data compression and caching policy in a WSN that can provably achieve the desired (maximal) difference between utility and cost.*

Contributions. In this paper, we propose a novel policy for data compression and caching in a WSN that provably optimizes utility and costs jointly. In this policy, data is compressed to a certain extent and cached at routers for a certain time period dictated by its TTL. To the best of our knowledge, we are the first to provide a theoretical basis for understanding the compression-caching tradeoff for data analytics in a WSN. In doing so, we provide analytical answers to how much compression should be performed at each router, and where the data should be cached in the network. Our contributions can be summarized as follows:

- To the best of our knowledge, the proposed joint data compression and caching framework is the first to provide a theoretical basis for understanding the tradeoff between data compression and caching in a WSN.
- We show how to jointly optimize utility and cost using this model. We achieve the optimal data compression ratio and caching location in a WSN. We propose a distributed algorithm that implements the optimal policy which can adapt to changes in the presence of limited information in the network (e.g., cache size and request processes).
- We evaluate our approach through extensive simulations on WSNs.

The rest of the paper is organized as follows. We present some technical preliminaries in Section II. We formulate the optimization problem in Section III, and propose an optimal distributed algorithm in Section IV. We present evaluation results in Section V and conclude in Section VII.

II. Preliminaries

We consider a WSN comprised a large number of sensors and routers. The end sensors usually generate data, and routers provide resources (such as caching and compression resources) on the route from end sensors to the sink node. We represent the WSN as a directed graph $G = (V, E)$. An illustrative example for data generation and request propagation in a tree-structured WSN is depicted in Figure 1.

A. TTL-Router

Each router has a cache to store data for compression. Denote by B_v the cache capacity at node $v \in V$. Let $\mathcal{K} \subset V$ be the set of end sensor nodes generating data with $|\mathcal{K}| = K$. Furthermore, we assume that each node j that receives data i from an end node k can compress it with a reduction ratio[1] $\delta_{ij}^{(k)}$, where $0 < \delta_{ij}^{(k)} \leq 1$, $\forall k, j$.

[1]defined as the ratio of the volume of the output data to the volume of input data at any node. We consider the compression that only reduces the quality of data (e.g. remove redundant information), but the total number of distinct data in the system remains the same.

Consider the cache at router j. Each data i is associated with a timer T_{ij}. When we focus on router j, we omit the subscript j. Consider the event when data i is requested. There are two cases: (i) if data i is not in the cache, data i is inserted into the cache and its timer is set to T_i; (ii) if data i is in the cache, its timer is reset to T_i. The timer decreases at a constant rate and the data is evicted once its timer expires.

B. Data Generation and Requests

We assume that sensor $k \in \mathcal{K}$ continuously generates data, which will be active for a time interval W and may be requested by analysts (users). If there is no request for this data in that time interval, the generated data becomes inactive and is discarded from the system. The generated data is compressed and cached along the path between the end sensor and sink node when a request is made for active data. Thus the total number of paths is $|\mathcal{K}| = K$, hence, w.l.o.g., $\mathcal{K}$ is also used to denote the set of all paths.

We consider TTL-routers in the WSN G, where each data has its own timer. Suppose data i is requested and routed along path p. There are two cases: (i) data i is not in any cache along path p, in which case data i is generated from the end sensor and inserted into the first TTL-router (denoted by 1)[2] on the path. Its timer is set to T_{i1}; (ii) if data i is in TTL-router l along path p, we consider the following simple strategy [3]

- **Move Copy Down with Push (MCDP)**: data i is moved to TTL-router $l+1$ preceding TTL-router l in which i is found, and the timer at TTL-router $l+1$ is set to $T_{i(l+1)}$. If timer T_{il} expires, data i is pushed one TTL-router back to TTL-router $l-1$ and the timer is set to $T_{i(l-1)}$.

C. Utility Function

Utility functions capture the satisfaction perceived by a user after being served a data. We associate each data $i \in \mathcal{D}$ with a utility function $U_i : [0, 1] \to \mathbb{R}$ that is a function of hit probability h_i. $U_i(\cdot)$ is assumed to be increasing, continuously differentiable, and strictly concave. In particular, for our numerical studies, we focus on the widely used β-fair utility functions [4] given by

$$U_i(h) = \begin{cases} w_i \frac{h^{1-\beta}}{1-\beta}, & \beta \geq 0, \beta \neq 1; \\ w_i \log h, & \beta = 1, \end{cases} \tag{1}$$

where $w_i > 0$ denotes a weight associated with data i.

III. Problem Formulation

In a WSN, each end sensor generates a sequence of data that analysts are interested in. Different end sensors may generate different types of data, i.e., there is no common data sharing between different end sensors.

W.l.o.g., we consider a particular end sensor k and denote the path from k to the sink as $p = (1, \cdots, |p|)$, where TTL-router $|p|$ is the sink node that serves the requests and TTL-router 1 is the end sensor that generates the data. Let the set

[2]Since we consider path p, for simplicity, we move the dependency on p and v, denote it as $1, \cdots, L$ directly.

of data generated by end sensor k be $\mathcal{D}^{(p)}$, where requests for data $i \in \mathcal{D}^{(p)}$ follow a Poisson process with rate λ_i.

Let $h_{ij}^{(p)}, T_{ij}^{(p)}$ denote the hit probability and TTL timer associated with data $i \in \mathcal{D}^{(p)}$ at node $j \in \{1, \cdots, |p|\}$, respectively. Let $\boldsymbol{h}_i^{(p)} = (h_{i1}^{(p)}, \cdots, h_{i|p|}^{(p)})$, $\boldsymbol{\delta}_i^{(p)} = (\delta_{i1}^{(p)}, \cdots, \delta_{i|p|}^{(p)})$ and $\boldsymbol{T}_i^{(p)} = (T_{i1}^{(p)}, \cdots, T_{i|p|}^{(p)})$. Let $\boldsymbol{h} = (\boldsymbol{h}_i^{(p)})$, $\boldsymbol{\delta} = (\boldsymbol{\delta}_i^{(p)})$ and $\boldsymbol{T} = (\boldsymbol{T}_i^{(p)})$ for $i \in \mathcal{D}^{(p)}$ and $p \in \mathcal{K}$.

A. Utilities

The overall utility for data i fetched over path p is

$$\sum_{j=1}^{|p|} \psi^{|p|-j} U_i^{(p)} \left(h_{ij}^{(p)} \prod_{l=1}^{j} \delta_{il}^{(p)} \right), \tag{2}$$

where $0 < \psi \leq 1$ is a discount factor capturing the data utility degradation along the request route. Here utilities not only capture hit probabilities but also characterize data quality degradation due to compression along the path.

B. Costs

We consider cost, for example delay, of routing the data along the path, which includes the cost to forward data to routers that caches it, the cost to search for the data along the path, and the cost to fetch cached data to analysts that sent the requests. Again, we assume that the per hop cost to transfer (search) data along the path is a function $c_f(\cdot)$ ($c_s(\cdot)$) of hit probabilities and compression ratios.

1) Forwarding Costs: Suppose a hit for data i occurs on TTL-router $j \in \{1, \cdots, |p|\}$, then the total cost to forward data i along p is

$$\sum_{j=1}^{|p|} \lambda_i \cdot j \cdot c_f \left(h_{ij}^{(p)} \prod_{l=1}^{j} \delta_{il}^{(p)} \right). \tag{3}$$

2) Search Costs: Given a hit for data i on TTL-router $j \in \{1, \cdots, |p|\}$, the total cost to search for data i along p is

$$\sum_{j=1}^{|p|} \lambda_i \cdot (|p| - j + 1) \cdot c_s(h_{ij}^{(p)}). \tag{4}$$

3) Fetching Costs: Upon a hit for data i on TTL-router $j \in \{1, \cdots, |p|\}$, the total cost to fetch data i along p is

$$\sum_{j=1}^{|p|} \lambda_i \cdot (|p| - j + 1) \cdot c_f \left(h_{ij}^{(p)} \prod_{l=1}^{j} \delta_{il}^{(p)} \right). \tag{5}$$

C. Hit Probability and Timer-based Policies

The mapping between hit probabilities and timers for different cache replication strategies was established in [5]. Using results from [5], we obtain the following expressions for corresponding data timers at a sensor j along path p.

$$T_{i1}^{(p)} = \frac{1}{\lambda_i} \log \left(1 + \frac{h_{i1}^{(p)}}{1 - \sum_{j \in \{1, \cdots, |p|\}} h_{ij}^{(p)}} \right), \tag{6a}$$

$$T_{ij}^{(p)} = \frac{1}{\lambda_i} \log \left(1 + \frac{h_{ij}^{(p)}}{h_{i(j-1)}^{(p)}} \right), \quad j = 2, \cdots, |p|. \tag{6b}$$

Note that

$$\sum_{j \in \{1, \cdots, |p|\}} h_{ij}^{(p)} \leq 1, \tag{7}$$

must hold during the mapping.

D. Optimization Formulation

Our objective is to determine a feasible TTL policy and compression ratio for data management in a WSN to maximize the difference between utilities and costs, i.e.,

$$\begin{aligned} F(\boldsymbol{h}, \boldsymbol{\delta}) = & \sum_{p \in \mathcal{K}} \sum_{i \in \mathcal{D}^{(p)}} \Bigg\{ \sum_{j=1}^{|p|} \psi^{|p|-j} U_i^{(p)} \left(h_{ij}^{(p)} \prod_{l=1}^{j} \delta_{il}^{(p)} \right) \\ & - \sum_{j=1}^{|p|} \lambda_i \cdot j \cdot c_f \left(h_{ij}^{(p)} \prod_{l=1}^{j} \delta_{il}^{(p)} \right) - \sum_{j=1}^{|p|} \lambda_i \cdot (|p| - j + 1) \cdot c_s(h_{ij}^{(p)}) \\ & - \sum_{j=1}^{|p|} \lambda_i \cdot (|p| - j + 1) \cdot c_f \left(h_{ij}^{(p)} \prod_{l=1}^{j} \delta_{il}^{(p)} \right) \Bigg\} \\ = & \sum_{p \in \mathcal{K}} \sum_{i \in \mathcal{D}^{(p)}} \Bigg\{ \sum_{j=1}^{|p|} \psi^{|p|-j} U_i^{(p)} \left(h_{ij}^{(p)} \prod_{l=1}^{j} \delta_{il}^{(p)} \right) \\ & - \Bigg[\sum_{j=1}^{|p|} \lambda_i (|p| + 1) c_f \left(h_{ij}^{(p)} \prod_{l=1}^{j} \delta_{il}^{(p)} \right) \\ & + \sum_{j=1}^{|p|} \lambda_i (|p| - j + 1) c_s(h_{ij}^{(p)}) \Bigg] \Bigg\}. \end{aligned} \tag{8}$$

Hence, the optimal TTL policy and compression ratio for MCDP should solve the following optimization problem:

$$\max \quad F(\boldsymbol{h}, \boldsymbol{\delta}) \tag{9a}$$

$$\text{s.t.} \quad \sum_{p: l \in p} \sum_{i \in \mathcal{D}^{(p)}} h_{il}^{(p)} \prod_{j=1}^{\mathcal{I}(l,p)} \delta_{ij}^{(p)} \leq B_l, \quad \forall l \in V, \tag{9b}$$

$$c_c \left(\sum_{i \in \mathcal{D}^{(p)}} \sum_{l=1}^{|p|} \prod_{j=1}^{l} \delta_{ij}^{(p)} \right) \leq O^{(p)}, \quad \forall p \in \mathcal{K}, \tag{9c}$$

$$\sum_{j \in \{1, \cdots, |p|\}} h_{ij}^{(p)} \leq 1, \quad \forall i \in \mathcal{D}^{(p)}, \forall p \in \mathcal{K}, \tag{9d}$$

$$0 \leq h_{ij}^{(p)} \leq 1, \quad \forall i \in \mathcal{D}^{(p)}, \forall p \in \mathcal{K}, j \in \{1, \cdots, |p|\}, \tag{9e}$$

$$0 < \delta_{ij}^{(p)} \leq 1, \quad \forall i \in \mathcal{D}^{(p)}, \forall p \in \mathcal{K}, j \in \{1, \cdots, |p|\}, \tag{9f}$$

where $\mathcal{I}(l, p)$ is the index of router j on path p and constraint (9c) is the energy available on path p to transmit the compressed data, and $c_c(\cdot)$ is the per unit energy consumption function for data transmission. Constraint (9d) is included in the formulation due to the mapping between hit probabilities and timers as discussed in Section III-C.

It is easy to check that (9) is a non-convex problem. In the following, we transform (9) into a convex one through Boyd's method (Section 4.5 [6]).

1) Convex Transformation: First, we define two new sets of variables for $i \in \mathcal{D}^{(p)}$, $l \in \{1, \cdots, |p|\}$ and $p \in \mathcal{K}$ as follows:

$$\log h_{ij}^{(p)} \triangleq \sigma_{ij}^{(p)}, \quad i.e., \quad h_{ij}^{(p)} = e^{\sigma_{ij}^{(p)}},$$
$$\log \delta_{ij}^{(p)} \triangleq \tau_{ij}^{(p)}, \quad i.e., \quad \delta_{ij}^{(p)} = e^{\tau_{ij}^{(p)}}, \tag{10}$$

and denote $\boldsymbol{\sigma}_i^{(p)} = (\sigma_{i1}^{(p)}, \cdots, \sigma_{ip}^{(p)})$, $\boldsymbol{\tau}_i^{(p)} = (\tau_{i1}^{(p)}, \cdots, \tau_{ip}^{(p)})$ and $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_i^{(p)})$, $\boldsymbol{\tau} = (\boldsymbol{\tau}_i^{(p)})$ for $i \in \mathcal{D}^{(p)}$ and $p \in \mathcal{K}$.

Then the objective function (8) can be transformed into

$$F(\boldsymbol{\sigma}, \boldsymbol{\tau}) = \sum_{p\in\mathcal{K}} \sum_{i\in\mathcal{D}^{(p)}} \Bigg\{ \sum_{j=1}^{|p|} \psi^{|p|-j} U_i^{(p)}\left(e^{\sigma_{ij}^{(p)} + \sum_{l=1}^{j} \tau_{il}^{(p)}}\right) - \Bigg[\sum_{j=1}^{|p|} \lambda_i(|p|+1) c_f\left(e^{\sigma_{ij}^{(p)} + \sum_{l=1}^{j} \tau_{il}^{(p)}}\right) + \sum_{j=1}^{|p|} \lambda_i(|p|-j+1) c_s\left(e^{\sigma_{ij}^{(p)}}\right)\Bigg]\Bigg\}. \tag{11}$$

We transform the constraints in a similar manner. Then we obtain the following transformed optimization problem

$$\max \quad F(\boldsymbol{\sigma}, \boldsymbol{\tau}) \tag{12a}$$
$$\text{s.t.} \quad \sum_{p: l\in p} \sum_{i\in\mathcal{D}^{(p)}} e^{\sigma_{il}^{(p)} + \sum_{j=1}^{\mathcal{I}(l,p)} \tau_{ij}^{(p)}} \le B_l, \quad \forall l \in V, \tag{12b}$$
$$c_c\left(\sum_{i\in\mathcal{D}^{(p)}} \sum_{l=1}^{|p|} e^{\sum_{j=1}^{l} \tau_{ij}^{(p)}}\right) \le O^{(p)}, \quad \forall p \in \mathcal{K}, \tag{12c}$$
$$\sum_{j\in\{1,\cdots,|p|\}} e^{\sigma_{ij}^{(p)}} \le 1, \quad \forall i \in \mathcal{D}^{(p)}, \forall p \in \mathcal{K}, \tag{12d}$$
$$\sigma_{ij}^{(p)} \le 0, \quad \forall i \in \mathcal{D}^{(p)}, \forall p \in \mathcal{K}, j \in \{1, \cdots, |p|\}, \tag{12e}$$
$$\tau_{ij}^{(p)} \le 0, \quad \forall i \in \mathcal{D}^{(p)}, \forall p \in \mathcal{K}, j \in \{1, \cdots, |p|\}, \tag{12f}$$

where $\mathcal{I}(l, p)$ is the index of router l on path p.

Lemma 1. *$U_i\left(e^{\sum_{k=1}^n x_k}\right)$ is a concave function for $\beta \ge 1$ where $U_i(\cdot)$ is defined in (1).*

Proof. We consider the following two cases, i.e., when $\beta = 1$ and $\beta \ne 1$.
Case 1($\beta = 1$): The utility function is $U_i(h) = w_i \log(h)$. Thus we have

$$U_i\left(e^{\sum_{k=1}^n x_k}\right) = w_i \log\left(e^{\sum_{k=1}^n x_k}\right) = w_i \sum_{k=1}^{n} x_k,$$

which is an affine function and thus concave as well.

Case 2($\beta \ne 1$): The utility function is $U_i(h) = w_i h^{1-\beta}/(1-\beta)$. Thus we have

$$U_i\left(e^{\sum_{k=1}^n x_k}\right) = w_i \frac{e^{(1-\beta)\sum_{k=1}^n x_k}}{1-\beta},$$

and the corresponding Hessian matrix is

$$H_i = (1-\beta) w_i e^{(1-\beta)\sum_{k=1}^n x_k} \begin{bmatrix} 1 & 1\cdots & 1 \\ 1 & 1\cdots & 1 \\ \vdots & \vdots & \vdots \\ 1 & 1\cdots & 1 \end{bmatrix}. \tag{13}$$

Note that, the unit matrix with all ones has eigenvalues n with multiplicity 1, and 0 with multiplicity $n-1$. The terms $e^{(1-\beta)\sum_{k=1}^n x_k}$ and w_i are always positive. Hence H_i is negative semi-definite, i.e., has non-positive eigenvalues, only when $1 - \beta < 0$. Combining both cases, $U_i\left(e^{\sum_{k=1}^n x_k}\right)$ is a concave function for $\beta \ge 1$. □

Theorem 1. *The transformed problem in (12) is convex $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$, when we consider the β-utility function with $\beta \ge 1$ and increasing convex cost functions $c_f(\cdot)$, $c_s(\cdot)$ and $c_c(\cdot)$.*

Proof. It is easy to check that the objective function in (12) is subject to convex inequality constraints. In particular, constraints (12e) and (12f) are affine convex functions. Inequality in constraint (12d) is convex due to convex composition under an affine mapping. Since the function $c_c(x)$ is convex and non-decreasing, by composition property (Section 3.2.4 [6]), constraint (12c) is also convex. Thus the feasible region in (12) is convex. A direct application of Lemma 1 yields the concavity condition for the objective $F(\boldsymbol{\sigma}, \boldsymbol{\tau})$. □

Theorem 2. *The optimization problems in (12) and (9) are equivalent.*

Proof. This is clear from the way we convexified the problem. □

IV. Distributed Algorithms

In Section III-D, we formulated a convex optimization problem with a fixed cache size. However, system parameters (e.g. cache size and request processes) can change over time, so it is not feasible to solve the optimization offline and implement the optimal strategy. Thus, we need to design distributed algorithms to implement the optimal strategy and adapt to the changes in the presence of limited information. In the following, we develop such an algorithm for MCDP.

A. Primal Algorithm

We aim to design an algorithm based on the optimization problem in (12), which is the primal formulation. We assume linear cost function with coefficient 1 for $c_c(\cdot)$. We first define the following objective function.

$$Z(\boldsymbol{\sigma}, \boldsymbol{\tau}) = F(\boldsymbol{\sigma}, \boldsymbol{\tau}) - \sum_{l\in V} C_B\left(\sum_{p: l\in p} \sum_{i\in\mathcal{D}^{(p)}} e^{\sigma_{il}^{(p)} + \sum_{j=1}^{\mathcal{I}(l,p)} \tau_{ij}^{(p)}} - B_l\right) - \sum_{p\in\mathcal{K}} C_O\left(\sum_{i\in\mathcal{D}^{(p)}} \sum_{l=1}^{|p|} e^{\sum_{j=1}^{l} \tau_{ij}^{(p)}} - O^{(p)}\right) - \sum_{p\in\mathcal{K}} \sum_{i\in\mathcal{D}^{(p)}} C_P\left(\sum_{j\in p} e^{\sigma_{ij}^{(p)}} - 1\right) - \sum_{p\in\mathcal{K}} \sum_{i\in\mathcal{D}^{(p)}} \sum_{j\in p} C_\tau\left(\tau_{ij}^{(p)}\right), \tag{14}$$

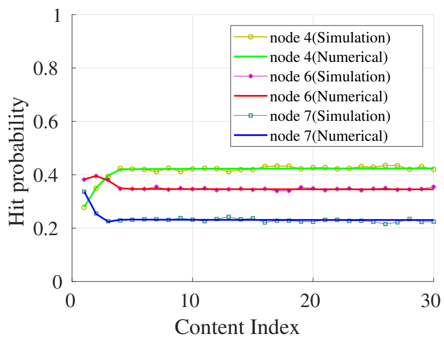

Fig. 2. Hit probability of MCDP under seven-node tree WSN.

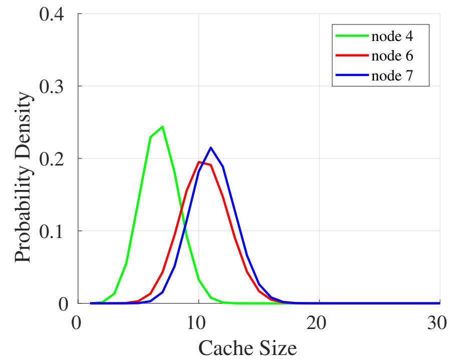

Fig. 3. Cache size of MCDP under seven-node tree WSN.

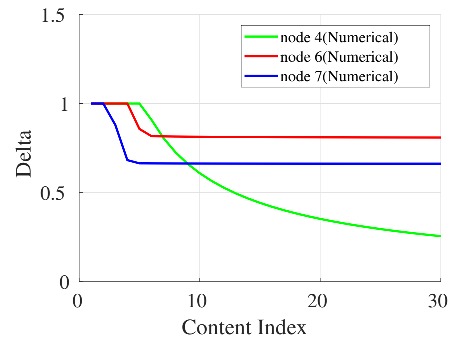

Fig. 4. Compression ratio of MCDP under a seven-node tree WSN.

where $C_B(\cdot)$, $C_O(\cdot)$, $C_P(\cdot)$ and $C_\tau(\cdot)$ are strictly convex and non-decreasing penalty functions denoting the cost for violating corresponding constraints (12b), (12c), (12d) and (12f) respectively. Using timer based caching techniques for controlling the hit probabilities with $0 \leq T_{il}^{(p)} \leq \infty$ ensures that constraint (12e) is always satisfied. So we ignore it in the formulation. It is clear that $Z(\cdot)$ is strictly concave. Hence, a natural way to obtain the maximum value of (14) is to use the standard *gradient ascent algorithm* to move the variable $\sigma_{il}^{(p)}$ and $\tau_{il}^{(p)}$ in the direction of the gradient, given as

$$\begin{aligned}\frac{\partial Z(\boldsymbol{\sigma},\boldsymbol{\tau})}{\partial \sigma_{il}^{(p)}} &= F'(\boldsymbol{\sigma},\boldsymbol{\tau}) - C_B'(B_{\text{curr},l} - B_l)\\ &- e^{\sigma_{il}^{(p)}} C_P'\left(\sum_{j\in p} e^{\sigma_{ij}^{(p)}} - 1\right)\\ &- e^{\sum_{j=1}^{l}\tau_{ij}^{(p)}} C_O'\left(\sum_{i\in\mathcal{D}^{(p)}}\sum_{l=1}^{|p|} e^{\sum_{j=1}^{l}\tau_{ij}^{(p)}} - O^{(p)}\right).\end{aligned} \tag{15}$$

Note that, $\sum_{p:l\in p}\sum_{i\in\mathcal{D}^{(p)}} e^{\sigma_{il}^{(p)} + \sum_{j=1}^{\mathcal{I}(l,p)}\tau_{ij}^{(p)}}$ basically represents the current size of cache l, also denoted as $B_{\text{curr},l}$. Also, $F'(\boldsymbol{\sigma},\boldsymbol{\tau}) = \partial F(\boldsymbol{\sigma},\boldsymbol{\tau})/\partial\sigma_{il}^{(p)}$. Similarly, we can evaluate $\partial Z(\boldsymbol{\sigma},\boldsymbol{\tau})/\partial\tau_{il}^{(p)}$. Furthermore, under TTL caching, the hit probabilities are controlled by the timers as discussed in Section III-C. Consequently we can update corresponding transformed variables through timers. Therefore, the primal algorithm for MCDP is given by

$$T_{il}^{(p)}[k] \leftarrow \begin{cases} \frac{1}{\lambda_i}\log\left(1 + \frac{e^{\sigma_{il}^{(p)}[k]}}{1-\left(e^{\sigma_{i1}^{(p)}[k]} + e^{\sigma_{i2}^{(p)}[k]} + \cdots + e^{\sigma_{i|p|}^{(p)}[k]}\right)}\right), \\ \qquad\qquad\qquad\qquad l = 1; \\ \frac{1}{\lambda_i}\log\left(1 + \frac{e^{\sigma_{il}^{(p)}[k]}}{e^{\sigma_{i(l-1)}^{(p)}[k]}}\right), \quad l = 2, \cdots, |p|, \end{cases} \tag{16a}$$

$$\sigma_{il}^{(p)}[k+1] \leftarrow \max\left\{0, \sigma_{il}^{(p)}[k] + \zeta_{il,\sigma}^{(p)}\frac{\partial Z(\boldsymbol{\sigma},\boldsymbol{\tau})}{\partial\sigma_{il}^{(p)}}\right\}, \tag{16b}$$

$$\tau_{il}^{(p)}[k+1] \leftarrow \max\left\{0, \tau_{il}^{(p)}[k] + \zeta_{il,\tau}^{(p)}\frac{\partial Z(\boldsymbol{\sigma},\boldsymbol{\tau})}{\partial\tau_{il}^{(p)}}\right\}, \tag{16c}$$

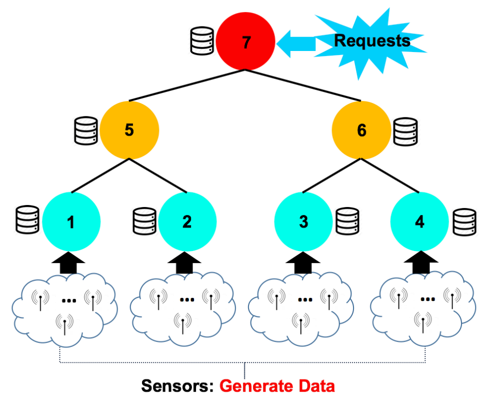

Fig. 5. A seven-node binary tree WSN.

where $\zeta_{il,\sigma}^{(p)}, \zeta_{il,\tau}^{(p)} > 0$ are the step-size parameters, and k is the iteration number incremented upon each request arrival.

Remark 1. *Note that, the primal formulation in* (16) *can be implemented distributively with respect to (w.r.t.) different data and paths by some amount of book-keeping and piggybacking. For example in* (15), $\sum_{j\in p} e^{\sigma_{ij}^{(p)}}$ *at router* l *can be computed by evaluating* $\sum_{j=1}^{l} e^{\sigma_{ij}^{(p)}}$ *and* $\sum_{j=l+1}^{|p|} e^{\sigma_{ij}^{(p)}}$ *during data forwarding and request propagation to router* l*, respectively.*

V. Evaluation

First, we consider a binary tree network with seven nodes as shown in Figure 5, where $\mathcal{K} = \{1, 2, 3, 4\}$. There are 4 leaf nodes, each is connected to 30 sensors. We assume that each sensor continuously generates content that are active for one time unit. Hence the paths are $p_1 = \{1, 5, 7\}$, $p_2 = \{2, 5, 7\}$, $p_3 = \{3, 6, 7\}$ and $p_4 = \{4, 6, 7\}$. Also let $B_v = 6$ for all leaf nodes $v \in \{1, \cdots, 4\}$, and $B_v = 10$ for nodes $v = 5, 6, 7$. Furthermore, for each leaf node, the content gathered from its sensors follows a Zipf distribution with parameters $\alpha_1 = 0.2$, $\alpha_2 = 0.4$, $\alpha_3 = 0.6$ and $\alpha_4 = 0.8$, respectively. For simplicity, we consider linear cost functions with coefficients 0.003 for $c_f(\cdot)$ and $c_s(\cdot)$, and 1 for $c_c(\cdot)$. The total energy constraint is set $O = 40$ for all paths. We consider the log utility function $U_i^{(k)}(x) = \lambda_i^{(k)}\log x$, where $\lambda_i^{(k)}$ is the request arrival rate for content i from sensor k. W.l.o.g., we assume the total arrival

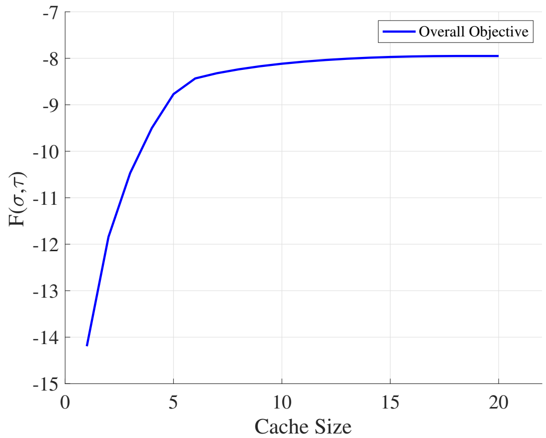

Fig. 6. Storage vs. overall objective under a seven-node tree WSN.

rate at each leaf node is 1, hence $\lambda_i^{(k)}$ equals to the content popularity.

Results for path p_4 are shown in Figures 2, 3 and 4. Again, we observe that our algorithm yields the exact optimal and empirical hit probabilities under MCDP for seven-node WSN. The density of number of content in the network concentrates around their corresponding cache sizes. Furthermore, we notice that the compression ratio δ at node 4 is much smaller than the ratios at nodes 6 and 7. Thus data compression happens at routers near to sensors so as to transmit less data long distances. This captures the trade-off between the costs of compression, communication and caching in our optimization framework. Similar observations can be made for the other three request paths and hence are omitted here.

We now focus on how cache capacity affects the overall objective $F(\sigma, \tau)$ as shown in Figure 6. With an increase in $B_l, \forall l \in V$, $F(\sigma, \tau)$ increases. Note that $F(\sigma, \tau)$ gradually converge to a value with increase in cache capacity. As $B_l \rightarrow |\mathcal{D}|$, $F(\sigma, \tau)$ becomes insensitive to B_l.

VI. Related Work

Data Compression. Compression is an important operation in analytics and has been studied in the past in various contexts [7]. In particular, for WSNs [8], [9], the goal is either to improve communication energy efficiency or energy tradeoff between communication, computation and caching. The goal of our work is different and is to achieve a desired difference between utility and cost.

TTL Caches. TTL caches have been employed in the Domain Name System (DNS) since the early days of Internet [10]. More recently, it has gained attention due to fact that a simple and tractable analysis can be modeled to mimic the behaviors of caching algorithms. [11], [12] first introduced the notion of characteristic time for LRU under IRM to show that TTL caches can be used to provide accurate estimates of the performance of large caches. It has been further generalized to other settings [5], [13], [14]. However, none of these works consider a joint optimization between data compression and caching in a WSN.

VII. Conclusion

We characterized the tradeoff among caching, compression and communication through our optimization framework by incorporating utilities of hit probability and costs of compression and communication. We identified the non-convexity issue and proposed a transformation technique to convert it into a convex problem. We also proposed a distributed algorithm which converges to the globally optimal solution. We showed the efficiency of our framework through numerical studies.

Acknowledgments

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.

References

[1] A. Vulimiri, C. Curino, P. B. Godfrey, T. Jungblut, K. Karanasos, J. Padhye, and G. Varghese, "Wanalytics: Geo-distributed Analytics for a Data Intensive World," in *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. ACM, 2015, pp. 1087–1092.

[2] F. Marcelloni and M. Vecchio, "An Efficient Lossless Compression Algorithm for Tiny Nodes of Monitoring Wireless Sensor Networks," *Comput. J.*, vol. 52, no. 8, pp. 969–987, 2009.

[3] I. Rodríguez, A. Ferragut, and F. Paganini, "Improving Performance of Multiple-level Cache Systems," in *SIGCOMM*, 2016.

[4] R. Srikant and L. Ying, *Communication Networks: an Optimization, Control, and Stochastic Networks Perspective*. Cambridge University Press, 2013.

[5] 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*, 2018.

[6] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[7] O. Boykin, S. Ritchie, I. O'Connell, and J. Lin, "Summingbird: A Framework for Integrating Batch and Online Mapreduce Computations," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1441–1451, 2014.

[8] 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.

[9] F. Zafari, J. Li, K. K. Leung, D. Towsley, and A. Swami, "Optimal energy tradeoff among communication, computation and caching with qoi-guarantee," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[10] J. Jung, A. Berger, and H. Balakrishnan, "Analysis of TTL-based Cache Networks," in *IEEE INFOCOM*, 2003.

[11] R. Fagin, "Asymptotic Miss Ratios over Independent References," *Journal of Computer and System Sciences*, vol. 14, no. 2, pp. 222–250, 1977.

[12] H. Che, Y. Tung, and Z. Wang, "Hierarchical Web Caching Systems: Modeling, Design and Experimental Results," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, pp. 1305–1314, 2002.

[13] D. S. Berger, P. Gland, S. Singla, and F. Ciucu, "Exact Analysis of TTL Cache Networks," *Performance Evaluation*, vol. 79, pp. 2–23, 2014.

[14] N. C. Fofack, P. Nain, G. Neglia, and D. Towsley, "Analysis of TTL-based Cache Networks," in *VALUETOOLS*, 2012.