Enhancing Energy Efficiency among Communication, Computation and Caching with QoI-Guarantee

Faheem Zafari[1], Jian Li[2], Kin K. Leung[1],
Don Towsley[2], Ananthram Swami[3]

[1]Imperial College London
[2]University of Massachusetts Amherst
[3]US Army Research Laboratory

Sept 25, 2017
DAIS-ITA Annual Fall Meeting 2017

Motivation

- Energy is one of the fundamental limitations of sensors

Motivation

- Energy is one of the fundamental limitations of sensors
- Sensors are responsible and consume energy for
  - **Communication**
  - **Computation**
  - **Caching**

Motivation

- Energy is one of the fundamental limitations of sensors
- Sensors are responsible and consume energy for
  - **Communication**
  - **Computation**
  - **Caching**
- Tradeoffs among communication, computation and caching energy costs
  - Computation (e.g. compression) consumes energy, but may reduce communication energy cost
  - Caching reduces communication cost but also consumes energy

The Big Question?

To achieve desirable tradeoffs

- How much data compression is needed?
- Where to cache data optimally?

Importance of C3 tradeoffs to DAIS-ITA/P1?

- Coalition infrastructure includes the C3 resources which we need to use efficiently

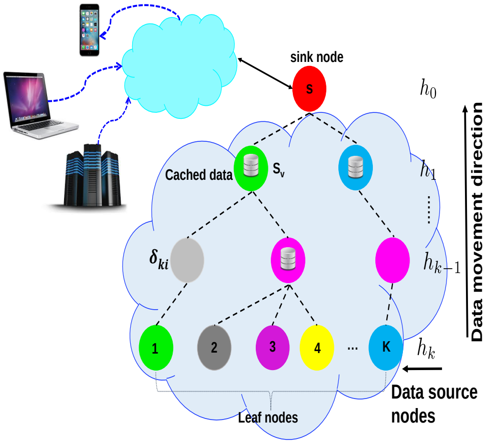
Importance of C3 tradeoffs to DAIS-ITA/P1?

- Coalition infrastructure includes the C3 resources which we need to use efficiently
- Sensor networks often have limited energy supply $\implies$ important to optimize energy usage

Importance of C3 tradeoffs to DAIS-ITA/P1?

- Coalition infrastructure includes the C3 resources which we need to use efficiently
- Sensor networks often have limited energy supply $\implies$ important to optimize energy usage
- We developed optimization solution for C3 tradeoffs that can be applied to other problems

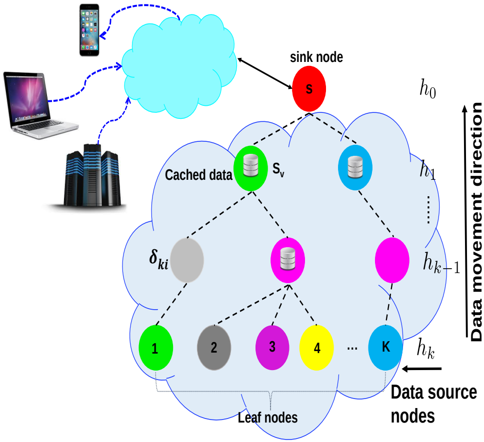
- Tree network is modeled as a directed graph $G = (V, E)$



Tree-Structured Network

System Model

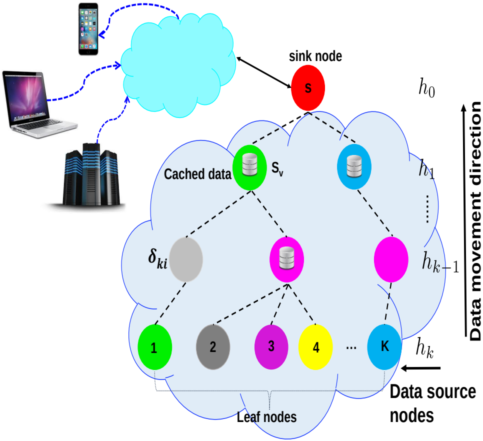
- Tree network is modeled as a directed graph $G = (V, E)$
- $\mathcal{K} \subseteq V$: the set of leaf nodes (data source) with $|\mathcal{K}| = K$



Tree-Structured Network

System Model

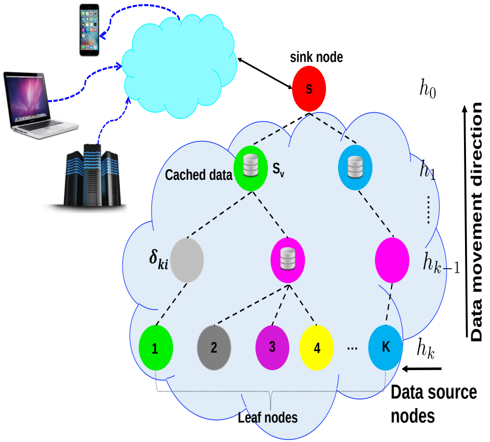
- Tree network is modeled as a directed graph $G = (V, E)$
- $\mathcal{K} \subseteq V$: the set of leaf nodes (data source) with $|\mathcal{K}| = K$
- Only leaf nodes $k \in \mathcal{K}$ can generate data of y_k bits



Tree-Structured Network

System Model

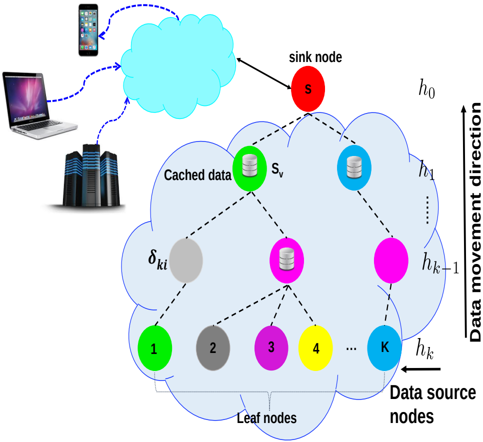
- Tree network is modeled as a directed graph $G = (V, E)$
- $\mathcal{K} \subseteq V$: the set of leaf nodes (data source) with $|\mathcal{K}| = K$
- Only leaf nodes $k \in \mathcal{K}$ can generate data of y_k bits
- Data is transmitted from a leaf node towards the sink node s



Tree-Structured Network

System Model

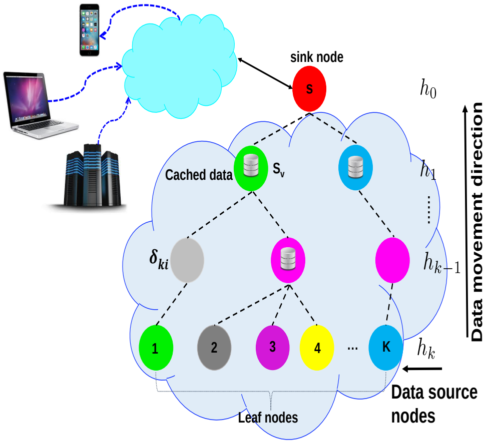
- Tree network is modeled as a directed graph $G = (V, E)$
- $\mathcal{K} \subseteq V$: the set of leaf nodes (data source) with $|\mathcal{K}| = K$
- Only leaf nodes $k \in \mathcal{K}$ can generate data of y_k bits
- Data is transmitted from a leaf node towards the sink node s
- Data of leaf node k can be compressed before further transmission with $\delta_{k,i}$ reduction factor



Tree-Structured Network

System Model

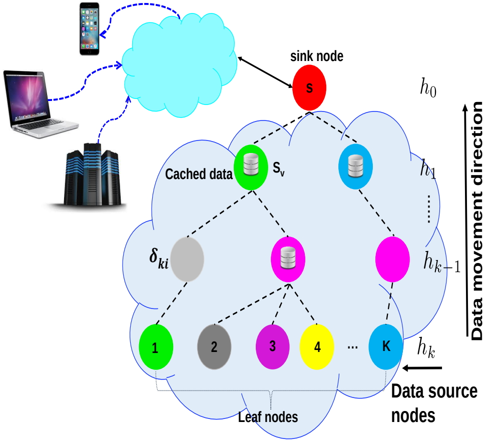
- Tree network is modeled as a directed graph $G = (V, E)$
- $\mathcal{K} \subseteq V$: the set of leaf nodes (data source) with $|\mathcal{K}| = K$
- Only leaf nodes $k \in \mathcal{K}$ can generate data of y_k bits
- Data is transmitted from a leaf node towards the sink node s
- Data of leaf node k can be compressed before further transmission with $\delta_{k,i}$ reduction factor
- Leaf node data can be cached at most in one node along the path towards sink node s



Tree-Structured Network

System Model

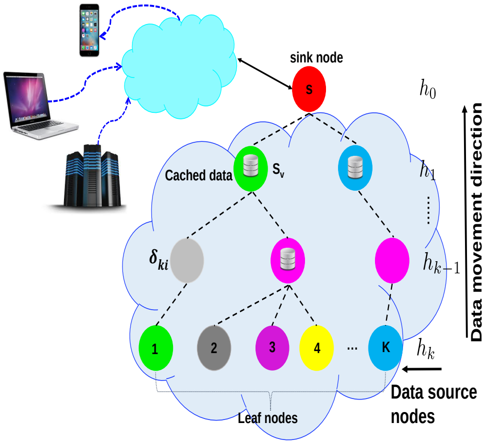
- The caching capacity at node $v \in V : S_v$



Tree-Structured Network

System Model

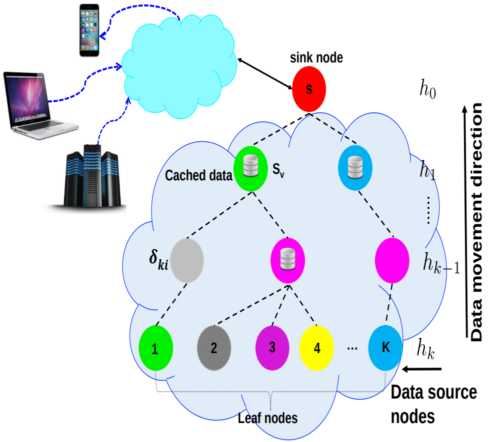
- The caching capacity at node $v \in V : S_v$
- Caching decision variable for leaf node data k at depth i: $b_{k,i} \in \{0, 1\}$



Tree-Structured Network

System Model

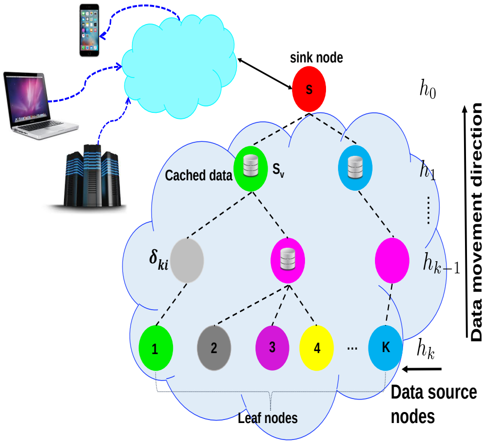
- The caching capacity at node $v \in V$: S_v
- Caching decision variable for leaf node data k at depth i: $b_{k,i} \in \{0, 1\}$
- Per-bit transmission, reception and computation energy cost: ε_{vT}, ε_{vR}, and ε_{vC}, define $f(\delta_v) = \varepsilon_{vR} + \varepsilon_{vT}\delta_{k,i} + \varepsilon_{kC}(\frac{1}{\delta_{k,i}} - 1)$



Tree-Structured Network

System Model

- The caching capacity at node $v \in V$: S_v
- Caching decision variable for leaf node data k at depth i: $b_{k,i} \in \{0, 1\}$
- Per-bit transmission, reception and computation energy cost: ε_{vT}, ε_{vR}, and ε_{vC}, define $f(\delta_v) = \varepsilon_{vR} + \varepsilon_{vT}\delta_{k,i} + \varepsilon_{kC}(\frac{1}{\delta_{k,i}} - 1)$
- During a time period T, R_k requests for data y_k generated by leaf node k



Tree-Structured Network

Energy Efficiency Optimization

- E_k^C: energy for data received, transmitted, and possibly compressed by all nodes on the path from leaf node k to sink node s

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

Energy Efficiency Optimization

- E_k^{C}: energy for data received, transmitted, and possibly compressed by all nodes on the path from leaf node k to sink node s

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

- E_k^{R} : the total energy consumed in responding to the subsequent $(R_k - 1)$ requests

$$E_k^{\mathrm{R}} = \sum_{i=0}^{h(k)} y_k (R_k - 1) \Big\{ f(\delta_{k,i}) \prod_{m=i+1}^{h(k)} \delta_{k,m} \Big(1 - \sum_{j=0}^{i-1} b_{k,j} \Big) + \Big(\prod_{m=i}^{h(k)} \delta_{k,m} \Big) b_{k,i} \Big(\frac{w_{ca} T}{(R_k - 1)} + \varepsilon_k T \Big) \Big\}. \tag{2}$$

$$E^{\text{total}}(\boldsymbol{\delta}, \boldsymbol{b}) \triangleq \sum_{k \in \mathcal{K}} \left(E_k^{\text{C}} + E_k^{\text{R}} \right) \qquad (3)$$

Non-convex Mixed Integer Nonlinear Programming (MINLP)

$$\begin{aligned}
\min_{\boldsymbol{\delta}, \boldsymbol{b}} \quad & E^{\textbf{total}}(\boldsymbol{\delta}, \boldsymbol{b}) \\
\textbf{s.t.} \quad & \sum_{k \in \mathcal{K}} y_k \prod_{i=0}^{h(k)} \delta_{k,i} \geq \gamma, \\
& b_{k,i} \in \{0, 1\}, \forall k \in \mathcal{K}, i = 0, \cdots, h(k), \\
& \sum_{k \in C_v} b_{k,h(v)} y_k \prod_{j=h(k)}^{h(v)} \delta_{k,j} \leq S_v, \forall\, v \in V, \\
& \sum_{i=0}^{h(k)} b_{k,i} \leq 1, \forall k \in \mathcal{K}. \qquad (4)
\end{aligned}$$

Part 1: Solving the Non-Convex MINLP Problem using our Variant of Spatial Branch and Bound Algorithm (V-SBB)

Symbolic Reformulation

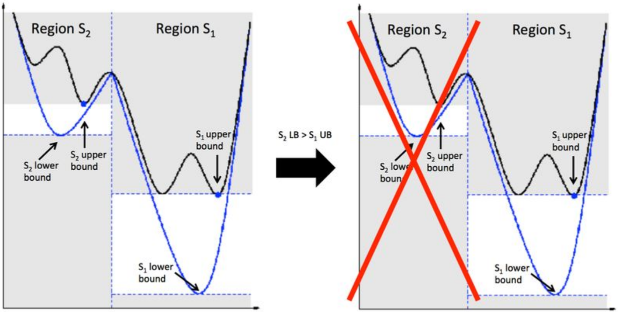
Non-Convex MINLP problem

$$\begin{aligned} \min \quad & \psi(X, Y) \\ \text{s.t.} \quad & G(X, Y) \leq 0 \\ & H(X, Y) = 0 \\ & X^L \leq X \leq X^U, X \in R \\ & Y \in [Y^L, \ldots, Y^U] \end{aligned}$$

Reformulated Problem

$$\begin{aligned} \min_{w} \quad & w_{obj} \\ \text{s.t.} \quad & Aw = b \\ & w^l \leq w \leq w^u \\ & Y \in [Y^L, \ldots, Y^U] \\ & w_k \equiv w_i w_j \quad \forall \quad (i, j, k) \in \tau_{bt} \\ & w_k \equiv \frac{w_i}{w_j} \quad \forall \quad (i, j, k) \in \tau_{lft} \\ & w_k \equiv w_i^n \quad \forall \quad (i, k, n) \in \tau_{et} \\ & w_k \equiv fn(w_i) \quad \forall \quad (i, k) \in \tau_{uft} \end{aligned}$$

Spatial Branch-and-Bound



BBM example (taken from `https://optimization.mccormick.northwestern.edu/index.php/File:SBB.png`)

V-SBB

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

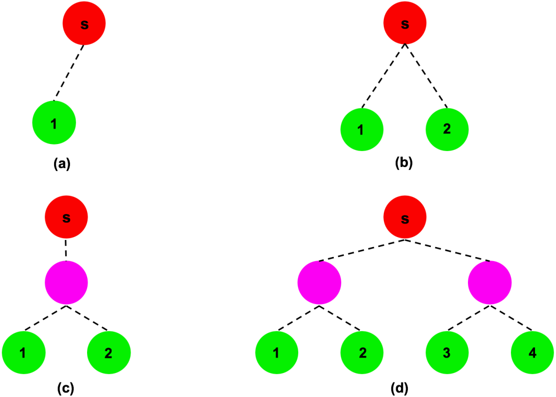
Step 1: Initialize $\phi^u := \infty$ and $\mathcal{L}$ to a single domain
Step 2: Choose a subregion $\mathcal{R} \in \mathcal{L}$ using *least lower bound rule*
if $\mathcal{L} = \emptyset$ or $\forall\ \mathcal{R} \in \mathcal{L}$, $\phi^{\mathcal{R},l}$ is infeasible **then** Go to Step 6
if $\phi^{\mathcal{R},l} \geq \phi^u - \epsilon$ **then** Go to Step 5
Step 3: Obtain the 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 ϕ^u is ϵ-global optimal

Decomposes non-linear functions of the original problem symbolically and recursively with simple operators into simple functions

Evaluation

Parameters used in simulations

Parameter	Value
y_k	1000
R_k	100
w_{ca}	1.88×10^{-6}
T	10s
ε_{vR}	50×10^{-9}
ε_{vT}	200×10^{-9}
ε_{cR}	80×10^{-9}
γ	$[1, \sum_{k \in \mathcal{K}} y_k]$



Candidate network topologies used in the experiments

Evaluations

The Best Solution to the Objective Function (Obj.) and Convergence time for Seven-node network (γ is the required QoI threshold)

Method	$\gamma = 1$		$\gamma = 1000$		$\gamma = 2000$		$\gamma = 3000$		$\gamma = 4000$	
	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	0.039	25.101	0.078	30.425	0.117	23.706	0.156	19.125

Summary of Results

- V-SBB outperforms all other algorithms in terms of obtaining better objective value
- Bonmin is faster but it has infeasibility issue and poor performance for some cases $\cdots$

Infeasibility: Not being able to find a solution when it exists

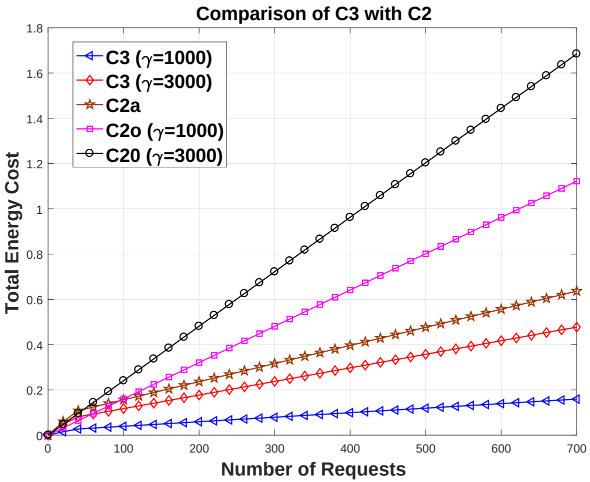
Infeasibility of Bonmin for different networks

Networks	(a)	(b)	(c)	(d)
Number of testing γ values	1000	2000	2000	4000
Number of infeasible solutions	0	0	1	216
Infeasibility (%)	0	0	0.05	5.4

Comparison between V-SBB and Bonmin for small γ values in seven-node network

Method	γ =1		γ =5		γ =50	
	Obj.	**Time (s)**	**Obj.**	**Time**	**Obj.**	**Time**
Bonmin	0.0002	0.214	0.0003	0.224	0.0021	0.364
V-SBB	0.00011	1871	0.00019	1243	0.0020	3325
Imp. (%)	52.45		50.30		4.62	

Importance of C3 over C2 tradeoffs



Comparison of C3 and C2 optimization for the seven nodes network.

Conclusions

- Formulated energy tradeoffs among communication, computation and caching with QoI guarantee as non-convex MINLP optimization problem

Conclusions

- Formulated energy tradeoffs among communication, computation and caching with QoI guarantee as non-convex MINLP optimization problem
- Proposed a variant of spatial branch-and-bound (V-SBB) algorithm, which can solve the MINLP with ϵ-optimality guarantee

Conclusions

- Formulated energy tradeoffs among communication, computation and caching with QoI guarantee as non-convex MINLP optimization problem
- Proposed a variant of spatial branch-and-bound (V-SBB) algorithm, which can solve the MINLP with ϵ-optimality guarantee
- Observed that C3 optimization improves energy efficiency by as much as 88% compared with either of the C2 optimizations

Ongoing Work

1. New formulation: minimize latency with energy constraints

Ongoing Work

1. New formulation: minimize latency with energy constraints
2. Design approximate algorithms to these non-convex MINLP problems to achieve a constant approximation ratio in polynomial time

Ongoing Work

1. New formulation: minimize latency with energy constraints
2. Design approximate algorithms to these non-convex MINLP problems to achieve a constant approximation ratio in polynomial time
3. Formulate Multi-Objective Optimization (MOO) for Software Defined Coalitions (SDC)

Ongoing Work

1. New formulation: minimize latency with energy constraints
2. Design approximate algorithms to these non-convex MINLP problems to achieve a constant approximation ratio in polynomial time
3. Formulate Multi-Objective Optimization (MOO) for Software Defined Coalitions (SDC)
    - Apply Cooperative Game Theory to coalition environment

Ongoing Work

1. New formulation: minimize latency with energy constraints
2. Design approximate algorithms to these non-convex MINLP problems to achieve a constant approximation ratio in polynomial time
3. Formulate Multi-Objective Optimization (MOO) for Software Defined Coalitions (SDC)
    - Apply Cooperative Game Theory to coalition environment
    - Explore Trust based regions

Thank you!

Backup Slides

System Model

- E_v : the total energy consumption at node v

$$E_v = E_{vR} + E_{vT} + E_{vC} + E_{vS}, \tag{5}$$

  - $E_{vR} = y_v \varepsilon_{vR}$ is the reception cost
  - $E_{vT} = y_v \varepsilon_{vT} \delta_v$ is the transmission cost
  - $E_{vC} = y_v \varepsilon_{vC} l_v(\delta_v)$ is the computation cost
  - $E_{vS} = w_{ca} y_v T$ is the storage cost
  - $l_v(\delta_v)$:a decreasing differentiable function of the reduction rate, e.g., $l_v(\delta_v) = \frac{1}{\delta_v} - 1$[1]
- During a time period of T, R_k requests for the data y_k generated by leaf node k

[1] Eswaran, Sharanya, et al. "Adaptive in-network processing for bandwidth and energy constrained mission-oriented multihop wireless networks." IEEE Transactions on Mobile Computing 11.9 (2012): 1484-1498.

Symbolic Reformulation

Example

We consider $k = 1$ and $h(k) = 1$ in (4), i.e., one leaf node and one sink node. Then (1) and (2) reduce to

$$
\begin{aligned}
E_1^C &= y_1 f(\delta_{1,0})\delta_{1,1} + y_1 f(\delta_{1,1}), \\
E_1^R &= y_1(R_1 - 1)\left[f(\delta_{1,0})\delta_{1,1} + \delta_{1,0}\delta_{1,1} b_{1,0}\left(\frac{w_{ca}T}{(R_1 - 1)} + \varepsilon_{1T}\right)\right] \\
&+ y_1(R_1 - 1)\left[f(\delta_{1,1})(1 - b_{1,0}) + \delta_{1,1} b_{1,1}\left(\frac{w_{ca}T}{(R_1 - 1)} + \varepsilon_{1T}\right)\right],
\end{aligned}
\tag{6}
$$

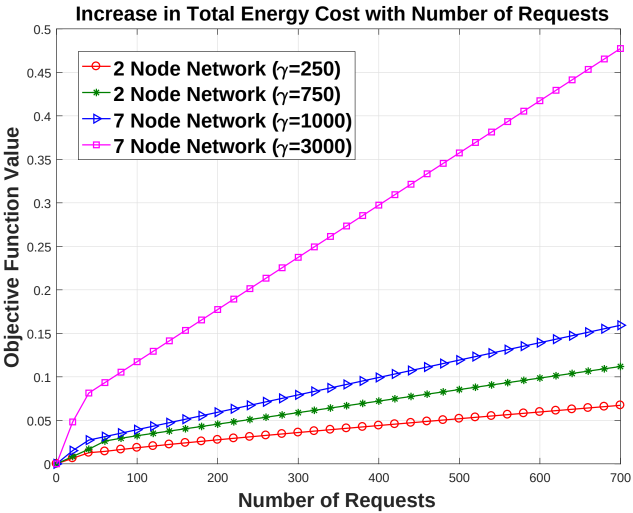
$$
\begin{aligned}
\min_{\boldsymbol{\delta}, \boldsymbol{b}} \quad & E^{\text{total}}(\boldsymbol{\delta}, \boldsymbol{b}) = E_1^C + E_1^R \\
\text{s.t.} \quad & y_1 \delta_{1,0}\delta_{1,1} \geq \gamma, \\
& b_{1,0}, b_{1,1} \in \{0, 1\}, \\
& b_{1,0} y_1 \delta_{1,0}\delta_{1,1} \leq S_0, \\
& b_{1,1} y_1 \delta_{1,1} \leq S_1, \\
& b_{1,0} + b_{1,1} \leq 1.
\end{aligned}
\tag{7}
$$

Symbolic Reformulation

$$
\begin{aligned}
\min_{\boldsymbol{\delta},\boldsymbol{b}} \quad & w_{\text{obj}} \\
\text{s.t.} \quad & y_1 w_{1,0}^{\text{bt}} \geq \gamma, \\
& b_{1,0}, b_{1,1} \in \{0,1\}, \\
& y_1 \overline{w}_{1,0}^{\text{bt}} \leq S_0, \\
& y_1 \tilde{w}_{1,1}^{\text{bt}} \leq S_1, \\
& b_{1,0} + b_{1,1} \leq 1, \\
& w_{1,0}^{\text{bt}} = \delta_{1,1} \times \delta_{1,0}, \\
& w_{1,0}^{\text{lft}} = \delta_{1,1} / \delta_{1,0}, \\
& \overline{w}_{1,0}^{\text{bt}} = b_{1,0} \times w_{1,0}^{b}, \\
& \tilde{w}_{1,1}^{\text{bt}} = b_{1,1} \times \delta_{1,1}, \\
& \tilde{w}_{1,0}^{\text{bt}} = \delta_{1,1} \times b_{1,0}, \\
& \tilde{w}_{1,0}^{\text{lft}} = b_{1,0} / \delta_{1,1},
\end{aligned}
$$

$$
\begin{aligned}
w_{\text{obj}} = {} & y_1 \varepsilon_{1R} \delta_{1,1} + \varepsilon_{1T} y_1 w_{1,0}^{\text{bt}} + y_1 \varepsilon_{1C} w_{1,0}^{\text{lft}} - y_1 \varepsilon_{1C} \delta_{1,1} \\
& + y_1 \varepsilon_{1R} + \varepsilon_{1T} y_1 \delta_{1,1} + y_1 \varepsilon_{1C} / \delta_{1,1} - y_1 \varepsilon_{1C} \\
& + y_1 (R_1 - 1) \Big[\varepsilon_{1R} \delta_{1,1} + \varepsilon_{1T} w_{1,0}^{\text{bt}} + \varepsilon_{1C} w_{1,0}^{\text{lft}} - \varepsilon_{1C} \delta_{1,1} \\
& + w_{ca} T \overline{w}_{1,0}^{bt} / (R_1 - 1) + \varepsilon_{1T} \overline{w}_{1,0}^{bt} \Big] + y_1 (R_1 - 1) \Big[\varepsilon_{1R} \\
& + \delta_{1,1} \varepsilon_{1T} + \varepsilon_{1C} / \delta_{1,1} - \varepsilon_{1C} - \varepsilon_{1R} b_{1,0} - \varepsilon_{1T} \tilde{w}_{1,0}^{bt} \\
& - \varepsilon_{1C} \tilde{w}_{1,0}^{lft} + \varepsilon_{1C} b_{1,0} + \tilde{w}_{1,1}^{bt} \Big(w_{ca} T / (R_1 - 1) + \varepsilon_{1T} \Big) \Big]
\end{aligned} \tag{8}
$$

Evaluations



Total Energy Costs vs. Number of Requests.