Energy-Latency Tradeoff for In-Network Function Computation in Random Networks

P. Balister[1] **B. Bollobás**[1] **A. Anandkumar**[2] **A.S. Willsky**[3]

[1]Dept. of Math., Univ. of Memphis, Memphis, TN, USA

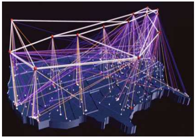
[2]Dept. of EECS, University of California, Irvine, CA, USA.

[3]Dept. of EECS, Massachusetts Institute of Technology, Cambridge, MA, USA.

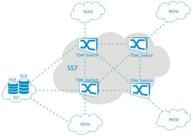
Presented by Dr. Ting He

IEEE INFOCOM 2011

In-network Function Computation



Internet

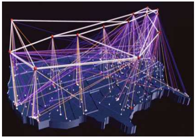


PSTN

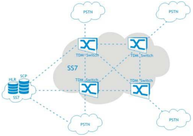
Traditional Wire-line Networks

- Over-provisioned links
- Layered architecture
- Data forwarding: no processing at intermediate nodes

In-network Function Computation



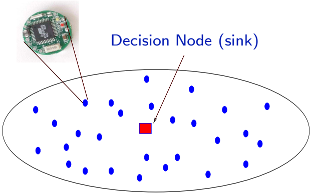
Internet



PSTN

Traditional Wire-line Networks

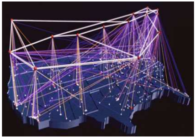
- Over-provisioned links
- Layered architecture
- Data forwarding: no processing at intermediate nodes



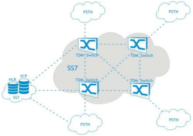
Energy-Constrained Sensor Networks

- Multihop wireless communication
- Transmission energy costs

In-network Function Computation



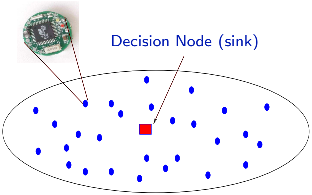
Internet



PSTN

Traditional Wire-line Networks

- Over-provisioned links
- Layered architecture
- Data forwarding: no processing at intermediate nodes



Energy-Constrained Sensor Networks

- Multihop wireless communication
- Transmission energy costs

In-network computation for energy savings

Energy-Latency Tradeoff for In-network Computation

Transmission Energy Costs for Wireless Communication

Cost for direct transmission between i and j scales as $R^{\nu}(i,j)$, where $2 \leq \nu \leq 6$ and ν is known as path-loss exponent.

Energy-Latency Tradeoff for In-network Computation

Transmission Energy Costs for Wireless Communication

Cost for direct transmission between i and j scales as $R^{\nu}(i,j)$, where $2 \leq \nu \leq 6$ and ν is known as path-loss exponent.

Achieving Energy Efficiency

- Multi-hop routing instead of direct transmission
- In-network computation to reduce amount of data transmitted

Energy-Latency Tradeoff for In-network Computation

Transmission Energy Costs for Wireless Communication

Cost for direct transmission between i and j scales as $R^{\nu}(i,j)$, where $2 \leq \nu \leq 6$ and ν is known as path-loss exponent.

Achieving Energy Efficiency

- Multi-hop routing instead of direct transmission
- In-network computation to reduce amount of data transmitted

Latency of Data Reception

Number of hops required for data transmission

Energy-Latency Tradeoff for In-network Computation

Transmission Energy Costs for Wireless Communication

Cost for direct transmission between i and j scales as $R^{\nu}(i,j)$, where $2 \leq \nu \leq 6$ and ν is known as path-loss exponent.

Achieving Energy Efficiency

- Multi-hop routing instead of direct transmission
- In-network computation to reduce amount of data transmitted

Latency of Data Reception

Number of hops required for data transmission

Energy-Latency Tradeoff

- Direct transmission: Higher cost but lower latency
- Multihop routing: Lower cost but higher latency

Problem Formulation

Goal

Design policy π to communicate certain function of data at nodes to the fusion center

Energy Consumption of a Policy π

Total energy costs $\sum_{(i,j)\in G_n^\pi} R^\nu(i,j)$

Latency of Function Computation

Delay for function value to reach fusion center

Optimal Energy-Latency Tradeoff

Minimize energy consumption subject to latency constraint

Can we design policies which achieve optimal energy-latency tradeoff?

Summary of Results

Stochastic Node Configuration

n nodes placed uniformly at random in $\mathbb{R}^d$ over area n

Summary of Results

Stochastic Node Configuration

n nodes placed uniformly at random in $\mathbb{R}^d$ over area n

Sum Function Computation

Deliver sum of data at nodes to fusion center

Summary of Results

Stochastic Node Configuration

n nodes placed uniformly at random in $\mathbb{R}^d$ over area n

Sum Function Computation

Deliver sum of data at nodes to fusion center

Energy-Latency Tradeoff for Sum Function Computation

- Propose novel policies which meet latency constraint
- Prove order-optimal energy-latency tradeoff
- Characterize scaling behavior with respect to path-loss exponent ν

Order-optimal Energy-Latency Tradeoff

Summary of Results Contd.,

Stochastic Node Configuration

n nodes placed uniformly at random in $\mathbb{R}^d$ over $[0, n^{1/d}]^d$

Clique-Based Function Computation

- Function which decomposes over cliques of a graph
- Relevant for statistical inference of graphical models (correlated sensor data)

Summary of Results Contd.,

Stochastic Node Configuration

n nodes placed uniformly at random in $\mathbb{R}^d$ over $[0, n^{1/d}]^d$

Clique-Based Function Computation

- Function which decomposes over cliques of a graph
- Relevant for statistical inference of graphical models (correlated sensor data)

Energy-Latency Tradeoff for Clique Function Computation

- Extend previous policy for this class of functions
- Prove order optimality under following conditions:
  1. Latency constraints belong to a certain range
  2. The graph governing the function is a proximity graph, e.g. k-nearest neighbor graph, random geometric graph

Related Work

Capacity of In-network Function Computation

- Rate of computation (Giridhar & Kumar 06)
- Single-shot computation considered here

Related Work

Capacity of In-network Function Computation

- Rate of computation (Giridhar & Kumar 06)
- Single-shot computation considered here

Minimum Broadcast Problem

- Minimize time of broadcast to all nodes from a single source (Ravi 94)
- Equivalent to latency of sum function computation
- Energy-latency tradeoff not considered before

Related Work

Capacity of In-network Function Computation

- Rate of computation (Giridhar & Kumar 06)
- Single-shot computation considered here

Minimum Broadcast Problem

- Minimize time of broadcast to all nodes from a single source (Ravi 94)
- Equivalent to latency of sum function computation
- Energy-latency tradeoff not considered before

Energy Optimization for Clique Function Computation

- Steiner-tree reduction (Anandkumar et. al. 08, 09)
- Order-optimality for random networks (Anandkumar et. al. 09)

Related Work

Capacity of In-network Function Computation

- Rate of computation (Giridhar & Kumar 06)
- Single-shot computation considered here

Minimum Broadcast Problem

- Minimize time of broadcast to all nodes from a single source (Ravi 94)
- Equivalent to latency of sum function computation
- Energy-latency tradeoff not considered before

Energy Optimization for Clique Function Computation

- Steiner-tree reduction (Anandkumar et. al. 08, 09)
- Order-optimality for random networks (Anandkumar et. al. 09)

Novelty: Energy-Latency Tradeoff for Function Computation

Outline

1 Introduction

2 Detailed Model and Formulation

3 Sum Function Computation

4 Conclusion

Detailed System Model

Communication Model

- Half-duplex nodes: no simultaneous transmission and reception
- Dedicated reception: Cannot receive data from multiple nodes
- No other interference constraints: orthogonal channels/directional antenna

Detailed System Model

Communication Model

- Half-duplex nodes: no simultaneous transmission and reception
- Dedicated reception: Cannot receive data from multiple nodes
- No other interference constraints: orthogonal channels/directional antenna

Propagation Model

- Unit transmission delay at all links

Stochastic Node Configuration $\mathbf{V}_n$

n nodes placed uniformly at random in $\mathbb{R}^d$ over $[0, n^{1/d}]^d$

Energy-Latency Tradeoff

Energy Consumption of a Policy π

$$\mathcal{E}^{\pi}(\mathbf{V}_n) := \sum_{(i,j)\in G_n^{\pi}} R^{\nu}(i,j)$$

Energy-Latency Tradeoff

Energy Consumption of a Policy π

$$\mathcal{E}^\pi(\mathbf{V}_n) := \sum_{(i,j)\in G_n^\pi} R^\nu(i,j)$$

Latency of Function Computation $L^\pi(\mathbf{V}_n)$

Delay for function value to reach fusion center

Minimum Latency

$$L^*(\mathbf{V}_n) := \min_\pi L^\pi(\mathbf{V}_n)$$

Optimal Energy-Latency Tradeoff

$$\mathcal{E}^*(\mathbf{V}_n;\delta) := \min_\pi \mathcal{E}^\pi(\mathbf{V}_n), \quad s.t.\ L^\pi \leq L^* + \delta.$$

Outline

1 Introduction

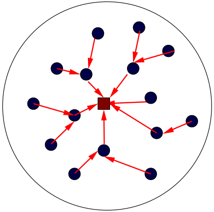
2 Detailed Model and Formulation

3 Sum Function Computation

4 Conclusion

Preliminaries for Sum Function Computation

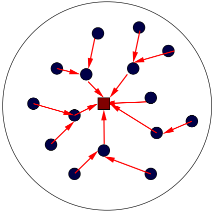
Computation Along a Tree T



- Links directed towards fusion center (root)
- Each node waits to receive data from children
- It then computes sum of values (along with own data) and forwards along outgoing link
- Process stops when data reaches fusion center

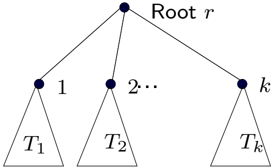
Preliminaries for Sum Function Computation

Computation Along a Tree T



- Links directed towards fusion center (root)
- Each node waits to receive data from children
- It then computes sum of values (along with own data) and forwards along outgoing link
- Process stops when data reaches fusion center

Latency Along a Tree



Latency L_T along tree T is

$$L_T = \max_{i=1,\ldots,k} \{i + L_{T_i}\}$$

- T_i: subtree rooted at node i
- $1, \ldots, k$: are of root such that $L_{T_1} \geq L_{T_2} \ldots \geq L_{T_k}$

Minimum Latency Tree

Minimum Latency Result

- Minimum latency for sum function computation over n nodes is $L^*(n) = \lceil \log_2 n \rceil$.
- $\iff$ max. # of nodes in tree with latency L is 2^L.

Minimum Latency Tree

Minimum Latency Result

- Minimum latency for sum function computation over n nodes is $L^*(n) = \lceil \log_2 n \rceil$.
- $\iff$ max. # of nodes in tree with latency L is 2^L.

Construction Minimum Latency Tree T^*

Recursively add child to each node already in tree

Minimum Latency Tree

Minimum Latency Result

- Minimum latency for sum function computation over n nodes is $L^*(n) = \lceil \log_2 n \rceil$.
- $\iff$ max. # of nodes in tree with latency L is 2^L.

Construction Minimum Latency Tree T^*

Recursively add child to each node already in tree

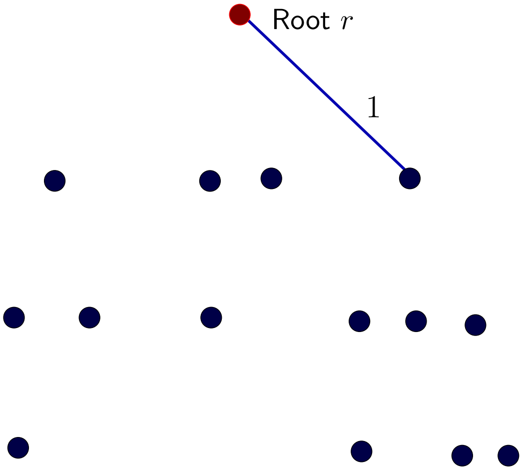
Level $l(e;T)$ of link e in tree T

$$l(e;T) = L_T - t_e.$$

- t_e: time of transmission at link e
- Process starts at time 0.

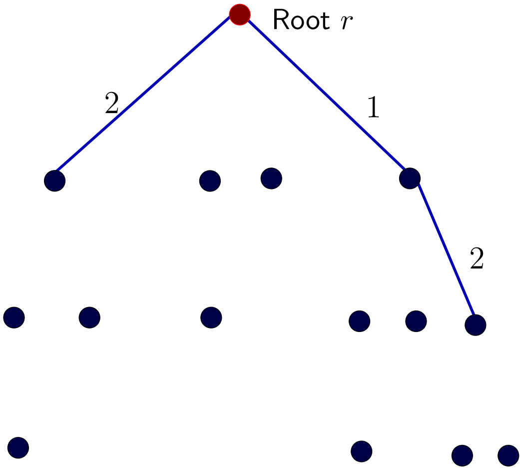
Example of Minimum Latency Tree T^*

Shown with edge-level labels



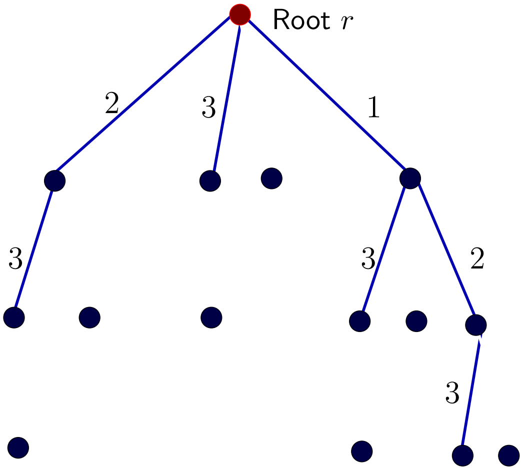
Example of Minimum Latency Tree T^*

Shown with edge-level labels



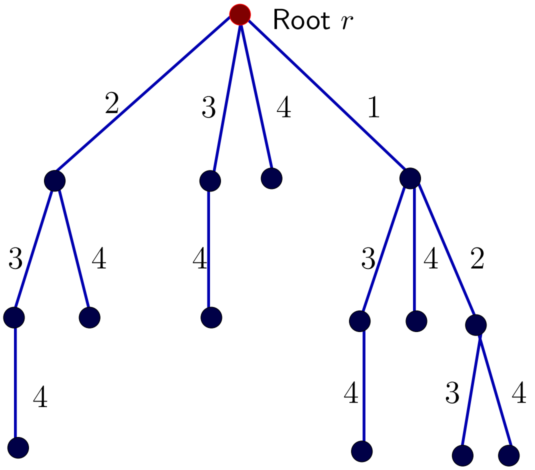
Example of Minimum Latency Tree T^*

Shown with edge-level labels



Example of Minimum Latency Tree T^*

Shown with edge-level labels



General Policy for Energy-Latency Tradeoff

Observations

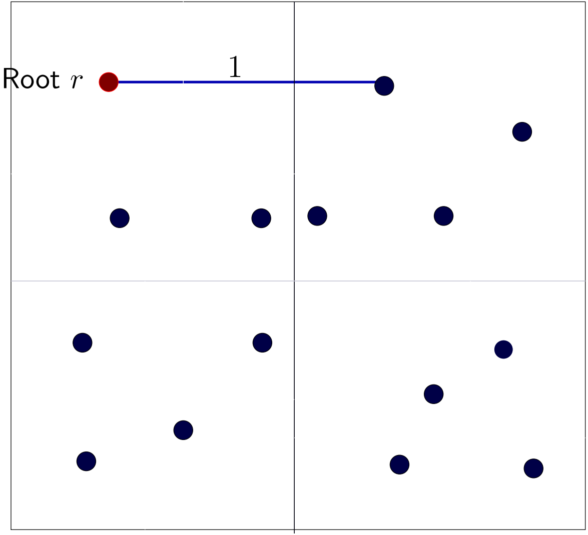
- Minimum Latency L^* independent of node locations $\mathbf{V}_n$
- Energy consumption depends on node locations $\mathbf{V}_n$

Construct aggregation tree T depending on $\mathbf{V}_n$

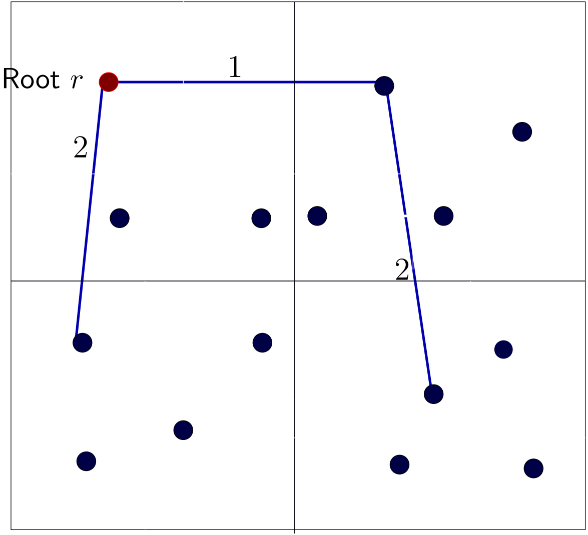
Overview of Algorithm π^{AGG}

- Iteratively bisect region under consideration
- Choose child in the other half
- Connect to the child along least energy route with at most w_k intermediate nodes

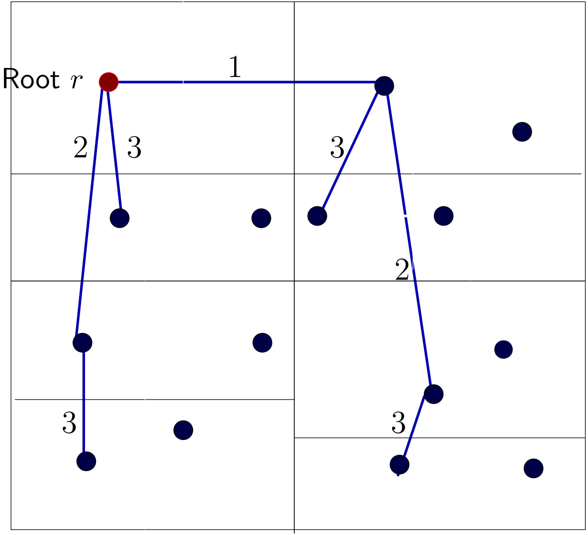
Example for π^{AGG} policy



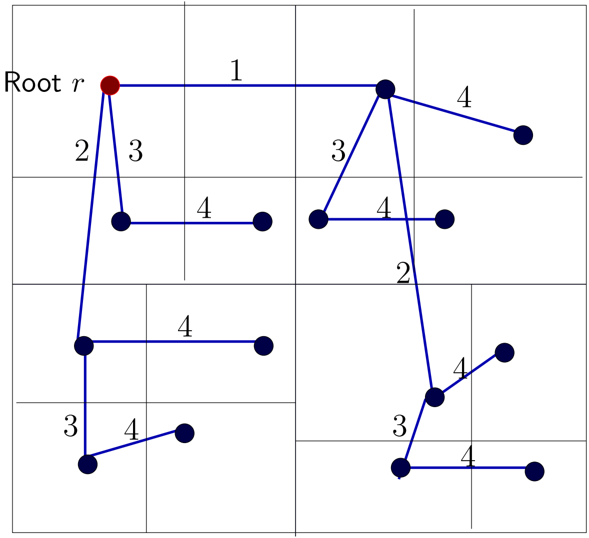
Example for π^{AGG} policy



Example for π^{AGG} policy



Example for π^{AGG} policy



Analysis of π^{AGG} policy

Latency under π^{AGG} policy

$$L^{\pi} = L^{*}(n) + \sum_{k=0}^{\lceil \log_2 n \rceil - 1} w_k$$

Analysis of π^{AGG} policy

Latency under π^{AGG} policy

$$L^{\pi} = L^{*}(n) + \sum_{k=0}^{\lceil \log_2 n \rceil - 1} w_k$$

Optimal Energy-Latency Tradeoff Problem

Minimize energy subject to latency constraint

$$\mathcal{E}^{*}(\mathbf{V}_n; \delta) := \min_{\pi} \mathcal{E}^{\pi}(\mathbf{V}_n), \quad s.t.\ L^{\pi} \le L^{*} + \delta.$$

Analysis of π^{AGG} policy

Latency under π^{AGG} policy

$$L^\pi = L^*(n) + \sum_{k=0}^{\lceil \log_2 n \rceil - 1} w_k$$

Optimal Energy-Latency Tradeoff Problem

Minimize energy subject to latency constraint

$$\mathcal{E}^*(\mathbf{V}_n; \delta) := \min_\pi \mathcal{E}^\pi(\mathbf{V}_n), \quad s.t.\ L^\pi \le L^* + \delta.$$

Choice of weights for π^{AGG} for optimal tradeoff

For $k = 0, \ldots, \lceil \log_2 n \rceil - 1$

$$w_k = \begin{cases} \lfloor \zeta \delta 2^{k(1/\nu - 1/d)} \rfloor & \text{if } \nu > d, \\ 0 & \text{o.w.} \end{cases}$$

Main Result: Optimal Energy-Latency Tradeoff

Optimal Energy-Latency Tradeoff

Minimize energy subject to latency constraint

$$\mathcal{E}^*(\mathbf{V}_n;\delta) := \min_{\pi} \mathcal{E}^{\pi}(\mathbf{V}_n), \quad s.t.\ L^{\pi} \leq L^* + \delta.$$

Theorem

For given δ, path-loss ν, dimension d, as number of nodes $n \to \infty$,

$$\mathbb{E}(\mathcal{E}^*(\mathbf{V}_n;\delta)) = \begin{cases} \Theta(n) & \nu < d, \\ O\Big(\max\{n, n(\log n)(1+\frac{\delta}{\log n})^{1-\nu}\}\Big) & \nu = d, \\ \Theta\Big(\max\{n, n^{\nu/d}(1+\delta)^{1-\nu}\}\Big) & \nu > d, \end{cases}$$

- Expectation is over node locations $\mathbf{V}_n$ of n
- Achieved by the policy π^{AGG}

Outline

1. Introduction
2. Detailed Model and Formulation
3. Sum Function Computation
4. Conclusion

Conclusion

Summary of Results

- Considered energy-latency tradeoff for function computation
- Considered sum function and function over cliques
- Proposed novel aggregation policies
- Proved order-optimal energy-latency tradeoff

Outlook

- Extensions beyond single-shot computation
- Multiple fusion centers with multiple functions for computation

Thank You !