A Framework for Designing Multihop Energy Harvesting Sensor Networks

Albert Sunny
Research supervisor: Prof. Joy Kuri

April 20, 2016

Department of Electronic Systems Engineering
Indian Institute of Science

Topics in my thesis

1. Managing TCP Transfers in IEEE 802.11 Infrastructure WLANs
2. Secure Transmission in Networks with Weak Eavesdroppers
3. A Framework for Designing Multihop Energy Harvesting Sensor Networks
4. Dynamics of History Dependent Epidemics in Temporal Networks

Topics in my thesis

1. Managing TCP Transfers in IEEE 802.11 Infrastructure WLANs
2. Secure Transmission in Networks with Weak Eavesdroppers
3. A Framework for Designing Multihop Energy Harvesting Sensor Networks
4. Dynamics of History Dependent Epidemics in Temporal Networks

Publications

1. Albert Sunny, Joy Kuri, "A Framework for Designing Multihop Energy Harvesting Sensor Networks," *to appear in IEEE Journal on Selected Areas in Communications - Second Issue on Green Communications and Networking*
2. Albert Sunny, Siddhartha Sarma, Joy Kuri, "Secure Transmission in Cooperative Networks with Weak Eavesdroppers," in *IEEE Signal Processing Letters, vol.22, no.10, pp.1693-1697, Oct. 2015*
3. Albert Sunny, Bhushan Kotnis, Joy Kuri, "Dynamics of History-dependent Epidemics in Temporal Networks," in *Physical Review E, vol.92, no.2, pp.022811-022820, Aug. 2015*
4. Albert Sunny, Siddhartha Sarma, Joy Kuri, "Beating Resource Constrained Eavesdroppers: A Physical Layer Security Study," in *WiOpt, pp.167-174, 25-29 May 2015*
5. Albert Sunny, Joy Kuri, "Link Dependence Probabilities in IEEE 802.11 Infrastructure WLANs," in *WiOpt, pp.148-153, 25-29 May 2015*

Motivation

Ingredients of multihop sensor networks

- **Sensor nodes:** equipped with multiple sensor modules, finite battery, finite storage, energy harvesting devices, and typically has a single antenna radio interface.

Ingredients of multihop sensor networks

- **Sensor nodes:** equipped with multiple sensor modules, finite battery, finite storage, energy harvesting devices, and typically has a single antenna radio interface.

- **Gateway nodes:** larger nodes equipped with a wireless interface for communications with the WSN, and a wired interface for communications with the controlling station.

Ingredients of multihop sensor networks

- **Sensor nodes:** equipped with multiple sensor modules, finite battery, finite storage, energy harvesting devices, and typically has a single antenna radio interface.
- **Gateway nodes:** larger nodes equipped with a wireless interface for communications with the WSN, and a wired interface for communications with the controlling station.
- **Adhoc architecture:** offer a range of benefits, including reliability, robustness, quick and easy network deployment, energy efficient network operations etc.

A few pre-deployment challenges

In most scenarios nodes are not mobile. Therefore, we need to plan the network in advance, for example,

- How many sensor nodes are needed?

A few pre-deployment challenges

In most scenarios nodes are not mobile. Therefore, we need to plan the network in advance, for example,

- How many sensor nodes are needed?
- How powerful should they be?

A few pre-deployment challenges

In most scenarios nodes are not mobile. Therefore, we need to plan the network in advance, for example,

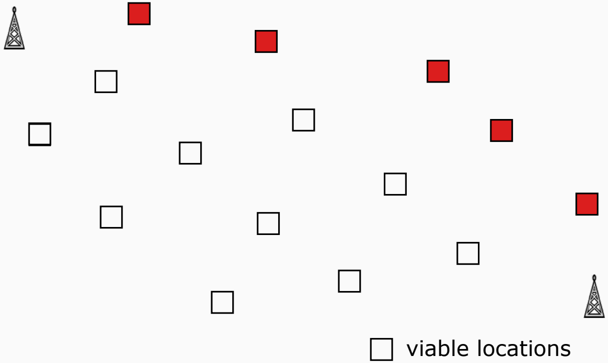
- How many sensor nodes are needed?
- How powerful should they be?
- How many gateway nodes are needed?

A few pre-deployment challenges

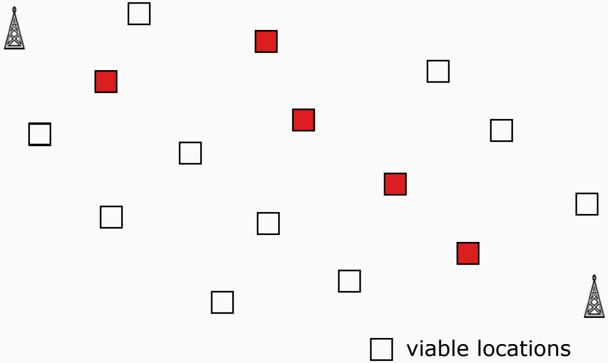
In most scenarios nodes are not mobile. Therefore, we need to plan the network in advance, for example,

- How many sensor nodes are needed?
- How powerful should they be?
- How many gateway nodes are needed?
- Where to locate the nodes?

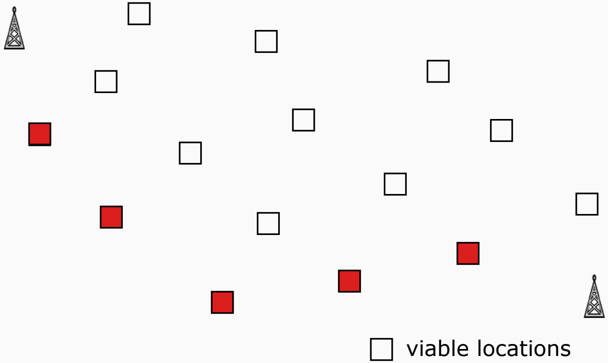
How to deploy?



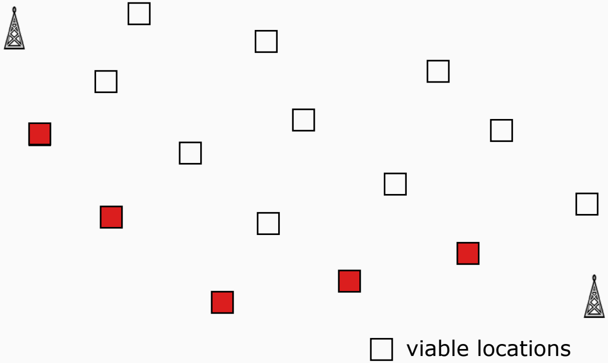
How to deploy?



How to deploy?



How to deploy?



System model

A utility function

- Time is divided into slots of length σ.

A utility function

- Time is divided into slots of length σ.
- $d_i(t)$: fraction of time sensor node is sensing the environment in the t^{th} slot.

A utility function

- Time is divided into slots of length σ.
- $d_i(t)$: fraction of time sensor node is sensing the environment in the t^{th} slot.
- Let us define $\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} d_i(t) = d_i$ - fraction of time sensor node i senses. We define the utility as $\sum_{i\in\mathcal{N}} U_i(d_i)$; U_is are increasing concave utility function.

A utility function

- Time is divided into slots of length σ.
- $d_i(t)$: fraction of time sensor node is sensing the environment in the t^{th} slot.
- Let us define $\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} d_i(t) = d_i$ - fraction of time sensor node i senses. We define the utility as $\sum_{i\in\mathcal{N}} U_i(d_i)$; U_is are increasing concave utility function.
- We use this utility function to compare and contrast different deployment scenarios

Evolution of the battery level

- Let $b_j(t)$ denote the battery level at node j, at the beginning of the t^{th} slot.
- Let $e_j^h(t)$ be the amount of energy harvested by node j in the t^{th} slot.
- e^s and e denote the energy consumed for sensing and active radio.
- b_{min} be the minimum battery level.
- $y_{kl}(t)$ part of flow from node k that is sent over link l in the t^{th} slot.

Then, the evolution of the battery level is given as follows:

$$b_j(t+1) = \min\Big\{b_{max}, b_j(t) + e_j^h(t) - e^s \cdot d_j(t) - \sum_{k \in \mathcal{N}} e \cdot \Big(\sum_{l \in \mathcal{O}(j)} y_{kl}(t) + \sum_{l \in \mathcal{I}(j)} y_{kl}(t) \Big) \Big\}$$

Evolution of the battery level

- Let $b_j(t)$ denote the battery level at node j, at the beginning of the t^{th} slot.
- Let $e_j^h(t)$ be the amount of energy harvested by node j in the t^{th} slot.
- e^s and e denote the energy consumed for sensing and active radio.
- b_{min} be the minimum battery level.
- $y_{kl}(t)$ part of flow from node k that is sent over link l in the t^{th} slot.

Then, the evolution of the battery level is given as follows:

$$b_j(t+1) = \min\Big\{ b_{max}, b_j(t) + e_j^h(t) - e^s \cdot d_j(t) - \sum_{k \in \mathcal{N}} e \cdot \Big(\sum_{l \in \mathcal{O}(j)} y_{kl}(t) + \sum_{l \in \mathcal{I}(j)} y_{kl}(t) \Big) \Big\}$$

Evolution of the battery level

- Let $b_j(t)$ denote the battery level at node j, at the beginning of the t^{th} slot.
- Let $e_j^h(t)$ be the amount of energy harvested by node j in the t^{th} slot.
- e^s and e denote the energy consumed for sensing and active radio.
- b_{min} be the minimum battery level.
- $y_{kl}(t)$ part of flow from node k that is sent over link l in the t^{th} slot.

Then, the evolution of the battery level is given as follows:

$$b_j(t+1) = \min\Big\{ b_{max}, b_j(t) + e_j^h(t) - e^s \cdot d_j(t) - \sum_{k \in \mathcal{N}} e \cdot \Big(\sum_{l \in \mathcal{O}(j)} y_{kl}(t) + \sum_{l \in \mathcal{I}(j)} y_{kl}(t) \Big) \Big\}$$

Evolution of the battery level

- Let $b_j(t)$ denote the battery level at node j, at the beginning of the t^{th} slot.
- Let $e_j^h(t)$ be the amount of energy harvested by node j in the t^{th} slot.
- e^s and e denote the energy consumed for sensing and active radio.
- b_{min} be the minimum battery level.
- $y_{kl}(t)$ part of flow from node k that is sent over link l in the t^{th} slot.

Then, the evolution of the battery level is given as follows:

$$b_j(t+1) = \min \Big\{ b_{max}, b_j(t) + e_j^h(t) - e^s \cdot d_j(t) - \sum_{k \in \mathcal{N}} e \cdot \Big(\sum_{l \in \mathcal{O}(j)} y_{kl}(t) + \sum_{l \in \mathcal{I}(j)} y_{kl}(t) \Big) \Big\}$$

Evolution of the data queue

- The nodes have a finite buffer size of q_{max} units.
- In the t^{th} slot, sensor node i produces $r^s \cdot d_i(t)$ units of data.
- The evolution of the data queue is given as follows:

$$q_i(t+1) = \min\Big\{q_{max}, q_i(t) + r^s \cdot d_i(t) + \sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{I}(k)} y_{kl}(t) - \sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{O}(k)} y_{kl}(t)\Big\}$$

Evolution of the data queue

- The nodes have a finite buffer size of q_{max} units.
- In the t^{th} slot, sensor node i produces $r^s \cdot d_i(t)$ units of data.
- The evolution of the data queue is given as follows:

$$q_i(t+1) = \min\Big\{q_{max}, q_i(t) + r^s \cdot d_i(t) + \sum_{k \in \mathcal{N}} \sum_{l \in \mathcal{I}(k)} y_{kl}(t) - \sum_{k \in \mathcal{N}} \sum_{l \in \mathcal{O}(k)} y_{kl}(t)\Big\}$$

Evolution of the data queue

- The nodes have a finite buffer size of q_{max} units.
- In the t^{th} slot, sensor node i produces $r^s \cdot d_i(t)$ units of data.
- The evolution of the data queue is given as follows:

$$q_i(t+1) = \min\Big\{q_{max}, q_i(t) + r^s \cdot d_i(t) + \sum_{k\in\mathcal{N}} \sum_{l\in\mathcal{I}(k)} y_{kl}(t) - \sum_{k\in\mathcal{N}} \sum_{l\in\mathcal{O}(k)} y_{kl}(t)\Big\}$$

Evolution of the data queue

- The nodes have a finite buffer size of q_{max} units.
- In the t^{th} slot, sensor node i produces $r^s \cdot d_i(t)$ units of data.
- The evolution of the data queue is given as follows:

$$q_i(t+1) = \min\Big\{q_{max}, q_i(t) + r^s \cdot d_i(t) + \sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{I}(k)} y_{kl}(t) - \sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{O}(k)} y_{kl}(t)\Big\}$$

Capacity and scheduling constraints

- Let $c_l(t)$ be the capacity of link l, in the t^{th} slot. Then, we have

$$\sum_{k \in \mathcal{N}} y_{kl}(t) \leq c_l(t)$$

Capacity and scheduling constraints

- Let $c_l(t)$ be the capacity of link l, in the t^{th} slot. Then, we have

$$\sum_{k \in \mathcal{N}} y_{kl}(t) \leq c_l(t)$$

- We have assumed the node-exclusive interference model. Conflicting links cannot be scheduled simultaneously. This can be captured using the notion of **maximal independent sets (MIS)**. Let $a_I(t)$ be the fraction of time MIS I is active, in the t^{th} slot. Then, we have

$$\sum_{I} a_I(t) \leq 1$$

Problem formulation

Long term time-averaged system

- In such WSN, the goal is to come up with optimal decision rules $\{\mathbf{d}(t), \mathbf{Y}(t), \mathbf{a}(t), t \geq 1\}$; usually posed as **Markov decision process (MDP)**.

Long term time-averaged system

- In such WSN, the goal is to come up with optimal decision rules $\{\mathbf{d}(t), \mathbf{Y}(t), \mathbf{a}(t), t \geq 1\}$; usually posed as **Markov decision process (MDP)**.
- However, in our setting, the reward depends on the long-term time-averaged quantities $\{d_1, d_2, \cdots, d_n\}$.

Long term time-averaged system

- In such WSN, the goal is to come up with optimal decision rules $\{\mathbf{d}(t), \mathbf{Y}(t), \mathbf{a}(t), t \geq 1\}$; usually posed as **Markov decision process (MDP)**.
- However, in our setting, the reward depends on the long-term time-averaged quantities $\{d_1, d_2, \cdots, d_n\}$.
- This enables us to look at the long-term time-averaged system.

Long term time-averaged system

- In such WSN, the goal is to come up with optimal decision rules $\{\mathbf{d}(t), \mathbf{Y}(t), \mathbf{a}(t), t \geq 1\}$; usually posed as **Markov decision process (MDP)**.
- However, in our setting, the reward depends on the long-term time-averaged quantities $\{d_1, d_2, \cdots, d_n\}$.
- This enables us to look at the long-term time-averaged system.
- It can be shown that the long-term time-averaged system under consideration should satisfy the following necessary condition

$$e^s \cdot d_j + \sum_{k \in \mathcal{N}} e \cdot \left(\sum_{l \in \mathcal{O}(j)} y_{kl} + \sum_{l \in \mathcal{I}(j)} y_{kl} \right) \leq e_j^h \, \forall j \in \mathcal{N} \qquad (1)$$

Long term time-averaged system

- In such WSN, the goal is to come up with optimal decision rules $\{\mathbf{d}(t), \mathbf{Y}(t), \mathbf{a}(t), t \geq 1\}$; usually posed as **Markov decision process (MDP)**.
- However, in our setting, the reward depends on the long-term time-averaged quantities $\{d_1, d_2, \cdots, d_n\}$.
- This enables us to look at the long-term time-averaged system.
- It can be shown that the long-term time-averaged system under consideration should satisfy the following necessary condition

$$e^s \cdot d_j + \sum_{k \in \mathcal{N}} e \cdot \Big(\sum_{l \in \mathcal{O}(j)} y_{kl} + \sum_{l \in \mathcal{I}(j)} y_{kl} \Big) \leq e_j^h \, \forall j \in \mathcal{N} \qquad (1)$$

- The above equation states that the rate of energy consumption should be less than or equal to the rate of energy harvesting.

Long term time-averaged system

$$\sum_{l \in \mathcal{O}(j)} y_{jl} = r^s \cdot d_j, \quad \sum_{l \in \mathcal{I}(j)} y_{jl} = 0 \quad \forall j \in \mathcal{N}$$

no accumulation at the sources

Long term time-averaged system

$$\sum_{l \in \mathcal{O}(j)} y_{jl} = r^s \cdot d_j, \quad \sum_{l \in \mathcal{I}(j)} y_{jl} = 0 \quad \forall j \in \mathcal{N}$$

no accumulation at the sources

$$\sum_{s \in \mathcal{S}} \sum_{l \in \mathcal{I}(s)} y_{jl} = r^s \cdot d_j \quad \forall j \in \mathcal{N}$$

no packet drops

Long term time-averaged system

$$\sum_{l \in \mathcal{O}(j)} y_{jl} = r^s \cdot d_j, \quad \sum_{l \in \mathcal{I}(j)} y_{jl} = 0 \quad \forall j \in \mathcal{N}$$

no accumulation at the sources

$$\sum_{s \in \mathcal{S}} \sum_{l \in \mathcal{I}(s)} y_{jl} = r^s \cdot d_j \quad \forall j \in \mathcal{N}$$

no packet drops

$$\sum_{l \in \mathcal{I}(j)} y_{kl} = \sum_{l \in \mathcal{O}(j)} y_{kl} \quad \forall j \in \mathcal{N}, \forall k \in \mathcal{N} \setminus \{j\}$$

flow conservation

Long term time-averaged system

$$\sum_{l \in \mathcal{O}(j)} y_{jl} = r^s \cdot d_j, \quad \sum_{l \in \mathcal{I}(j)} y_{jl} = 0 \quad \forall j \in \mathcal{N}$$

no accumulation at the sources

$$\sum_{s \in \mathcal{S}} \sum_{l \in \mathcal{I}(s)} y_{jl} = r^s \cdot d_j \quad \forall j \in \mathcal{N}$$

no packet drops

$$\sum_{l \in \mathcal{I}(j)} y_{kl} = \sum_{l \in \mathcal{O}(j)} y_{kl} \quad \forall j \in \mathcal{N}, \forall k \in \mathcal{N} \setminus \{j\}$$

flow conservation

$$\sum_{k \in \mathcal{N}} y_{kl} \leq (\mathbf{M} \cdot \mathbf{a})_l$$

rate of flow on link $\leq$ effective link capacity

Long term time-averaged system

$$\sum_{s \in \mathcal{S}} \sum_{l \in \mathcal{I}(s)} y_{jl} = r^s \cdot d_j \quad \forall j \in \mathcal{N}$$

no packet drops

$$\sum_{l \in \mathcal{I}(j)} y_{kl} = \sum_{l \in \mathcal{O}(j)} y_{kl} \quad \forall j \in \mathcal{N}, \forall k \in \mathcal{N} \setminus \{j\}$$

flow conservation

$$\sum_{k \in \mathcal{N}} y_{kl} \leq (\mathbf{M} \cdot \mathbf{a})_l$$

rate of flow on link $\leq$ effective link capacity

$$\sum_{l} a_l \leq 1$$

two different MIS cannot be active simultaneously

An optimization problem

$$P_1: \quad \max_{\{\mathbf{a}\geq\mathbf{0}, \quad \mathbf{Y}\geq\mathbf{0}, \quad \mathbf{d}\in[0,1]^{|\mathcal{N}|}\}} \quad \sum_{j\in\mathcal{N}} U_j(d_j)$$

Subject to:

$$\sum_{l\in\mathcal{O}(j)} y_{jl} = r^s \cdot d_j, \quad \sum_{l\in\mathcal{I}(j)} y_{jl} = 0 \quad \forall j \in \mathcal{N} \tag{2}$$

$$\sum_{s\in\mathcal{S}} \sum_{l\in\mathcal{I}(s)} y_{jl} = r^s \cdot d_j \quad \forall j \in \mathcal{N} \tag{3}$$

$$\sum_{l\in\mathcal{I}(j)} y_{kl} = \sum_{l\in\mathcal{O}(j)} y_{kl} \quad \forall j \in \mathcal{N}, \forall k \in \mathcal{N} \setminus \{j\} \tag{4}$$

$$e^s \cdot d_j + \sum_{k\in\mathcal{N}} e \cdot \Big(\sum_{l\in\mathcal{O}(j)} y_{kl} + \sum_{l\in\mathcal{I}(j)} y_{kl} \Big) \leq e_j^h \, \forall j \in \mathcal{N} \tag{5}$$

$$\sum_{k\in\mathcal{N}} y_{kl} \leq (\mathbf{M} \cdot \mathbf{a})_l \tag{6}$$

$$\sum_{l} a_l \leq 1 \tag{7}$$

An alternate formulation

How to avoid computing matrix M?

- Replace the MIS constraints with the following clique constraints.

$$\mathbf{Fc} \leq \mathbf{1}$$

where **F** is the contention matrix.

How to avoid computing matrix M?

- Replace the MIS constraints with the following clique constraints.

$$\mathbf{F}\mathbf{c} \leq \mathbf{1}$$

  where $\mathbf{F}$ is the contention matrix.
- For the node-exclusive interference model, the clique constraints can be written as

$$\sum_{l \in \mathcal{I}(j) \cup \mathcal{O}(j)} \frac{\sum_{k \in \mathcal{N}} y_{kl}(t)}{c_l^0} \leq 1 \quad \forall j \in \mathcal{N} \tag{8}$$

How to avoid computing matrix M?

- Replace the MIS constraints with the following clique constraints.

$$\mathbf{Fc} \leq \mathbf{1}$$

  where $\mathbf{F}$ is the contention matrix.

- For the node-exclusive interference model, the clique constraints can be written as

$$\sum_{l \in \mathcal{I}(j) \cup \mathcal{O}(j)} \frac{\sum_{k \in \mathcal{N}} y_{kl}(t)}{c_l^0} \leq 1 \quad \forall j \in \mathcal{N} \tag{8}$$

- We note that for the node-exclusive interference model, **F** has a computational complexity of $O(|\mathcal{L}|)$. Clique constraints are computationally scalable. However, they are necessary but not sufficient.

An alternate optimization problem

$$P_2 : \quad \max_{\{\mathbf{a}\geq\mathbf{0}, \quad \mathbf{Y}\geq\mathbf{0}, \quad \mathbf{d}\in[0,1]^{|\mathcal{N}|}\}} \quad \sum_{j\in\mathcal{N}} U_j(d_j)$$

Subject to:

$$\sum_{l\in\mathcal{O}(j)} y_{jl} = r^s \cdot d_j, \quad \sum_{l\in\mathcal{I}(j)} y_{jl} = 0 \quad \forall j \in \mathcal{N} \tag{9}$$

$$\sum_{s\in\mathcal{S}} \sum_{l\in\mathcal{I}(s)} y_{jl} = r^s \cdot d_j \quad \forall j \in \mathcal{N} \tag{10}$$

$$\sum_{l\in\mathcal{I}(j)} y_{kl} = \sum_{l\in\mathcal{O}(j)} y_{kl} \quad \forall j \in \mathcal{N}, \forall k \in \mathcal{N} \setminus \{j\} \tag{11}$$

$$e^s \cdot d_j + \sum_{k\in\mathcal{N}} e \cdot \Big(\sum_{l\in\mathcal{O}(j)} y_{kl} + \sum_{l\in\mathcal{I}(j)} y_{kl} \Big) \leq e_j^h \, \forall j \in \mathcal{N} \tag{12}$$

$$\mathbf{Fc} \leq \mathbf{1} \tag{13}$$

$$\sum_{l} a_l \leq 1 \tag{14}$$

A solution approach

- After replacing the MIS constraint with computationally tractable Clique constraints, we obtain a new optimization problem (problem P_2).

A solution approach

- After replacing the MIS constraint with computationally tractable Clique constraints, we obtain a new optimization problem (problem P_2).
- Problem P_2 satisfies *Slater's condition*. Therefore, we can solve it by solving its dual obtained by relaxing the energy and the capacity constraints.

$$\min_{\boldsymbol{\beta} \geq \mathbf{0}, \boldsymbol{\gamma} \geq \mathbf{0}} D(\boldsymbol{\beta}, \boldsymbol{\gamma}) \tag{15}$$

where

$$D(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \max_{\mathbf{d}, \mathbf{Y}, \mathbf{c}} \Bigg\{ \sum_{j \in \mathcal{N}} \Bigg(U_j(d_j) + \beta_j \cdot \Bigg(e_j - e^s d_j - \sum_{k \in \mathcal{N}} e \cdot \Big(\sum_{l \in \mathcal{O}(j)} y_{kl} + \sum_{l \in \mathcal{I}(j)} y_{kl} \Big) \Bigg) \Bigg) + \sum_{l \in \mathcal{L}} \gamma_l \Big(c_l - \sum_{k \in \mathcal{N}} y_{kl} \Big) \Bigg\} \tag{16}$$

Subject to: flow and clique constraints, $\mathbf{c} \geq \mathbf{0}, \quad \mathbf{Y} \geq \mathbf{0}, \quad \mathbf{d} \in [0,1]^{|\mathcal{N}|}$

A solution approach

- After replacing the MIS constraint with computationally tractable Clique constraints, we obtain a new optimization problem (problem P_2).
- Problem P_2 satisfies *Slater's condition*. Therefore, we can solve it by solving its dual obtained by relaxing the energy and the capacity constraints.

$$\min_{\boldsymbol{\beta} \geq \mathbf{0}, \boldsymbol{\gamma} \geq \mathbf{0}} D(\boldsymbol{\beta}, \boldsymbol{\gamma}) \tag{15}$$

where

$$D(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \max_{\mathbf{d}, \mathbf{Y}, \mathbf{c}} \Bigg\{ \sum_{j \in \mathcal{N}} \Big(U_j(d_j) + \beta_j \cdot \Big(e_j - e^s d_j - \sum_{k \in \mathcal{N}} e \cdot \Big(\sum_{l \in \mathcal{O}(j)} y_{kl} + \sum_{l \in \mathcal{I}(j)} y_{kl} \Big) \Big) \Big) + \sum_{l \in \mathcal{L}} \gamma_l \Big(c_l - \sum_{k \in \mathcal{N}} y_{kl} \Big) \Bigg\} \tag{16}$$

Subject to: flow and clique constraints, $\mathbf{c} \geq \mathbf{0}$, $\mathbf{Y} \geq \mathbf{0}$, $\mathbf{d} \in [0,1]^{|\mathcal{N}|}$

- The Lagrange multipliers in the dual can be interpreted as prices.

A solution approach

- Following standard approaches, this dual can be decomposed into two sub-problems that can be solved independent of each other.

A solution approach

- Following standard approaches, this dual can be decomposed into two sub-problems that can be solved independent of each other.
- Scheduling subproblem

$$\max_{\mathbf{c}\geq\mathbf{0}} \boldsymbol{\gamma}^T\mathbf{c} \quad \text{subject to} \quad \mathbf{F}\mathbf{c} \leq \mathbf{1}$$

A solution approach

- Following standard approaches, this dual can be decomposed into two sub-problems that can be solved independent of each other.
- Scheduling subproblem

$$\max_{\mathbf{c} \geq \mathbf{0}} \boldsymbol{\gamma}^T \mathbf{c} \quad \text{subject to} \quad \mathbf{F}\mathbf{c} \leq \mathbf{1}$$

- Joint sensing fraction allocation and routing subproblem

$$\max_{\mathbf{d},\mathbf{Y}} \left\{ \sum_{j \in \mathcal{N}} \left(U_j(d_j) - \beta_j e^s d_j \right) - \sum_{k \in \mathcal{N}} \sum_{l \in \mathcal{L}} \gamma_l \cdot y_{kl} - \sum_{k \in \mathcal{N}} \left(\sum_{j \in \mathcal{N}} \beta_j \cdot e \cdot \left(\sum_{l \in \mathcal{O}(j)} y_{kl} + \sum_{l \in \mathcal{I}(j)} y_{kl} \right) \right) \right\} \quad (17)$$

Subject to: $\mathbf{Y} \geq \mathbf{0}, \mathbf{d} \in [0,1]^{|\mathcal{N}|}$ and flow conservation equations

A solution approach

- Joint sensing fraction allocation and routing subproblem

$$d_j(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \left[U'^{-1} \left(\beta_j \cdot e^s + r^s \cdot c_j^{lcp}(\boldsymbol{\beta}, \boldsymbol{\gamma}) \right) \right]^*$$

where c_j^{lcp} is the cost of least-cost path and is given as

$$c_j^{lcp} = \arg\min_{s \in \mathcal{S}} \min_{P \in \mathcal{P}_{js}} \left(\sum_{l \in P \cap \mathcal{L}} \gamma_l + 2e \cdot \sum_{k \in P \cap \mathcal{N}} \beta_k \right)$$

A solution approach

- Joint sensing fraction allocation and routing subproblem

$$d_j(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \left[{U'}^{-1} \left(\beta_j \cdot e^s + r^s \cdot c_j^{lcp}(\boldsymbol{\beta}, \boldsymbol{\gamma}) \right) \right]^*$$

where c_j^{lcp} is the cost of least-cost path and is given as

$$c_j^{lcp} = \arg \min_{s \in \mathcal{S}} \min_{P \in \mathcal{P}_{js}} \left(\sum_{l \in P \cap \mathcal{L}} \gamma_l + 2e \cdot \sum_{k \in P \cap \mathcal{N}} \beta_k \right)$$

- Scheduling subproblem can be solved using linear programming.

A solution approach

- Joint sensing fraction allocation and routing subproblem

$$d_j(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \left[{U'}^{-1} \left(\beta_j \cdot e^s + r^s \cdot c_j^{lcp}(\boldsymbol{\beta}, \boldsymbol{\gamma}) \right) \right]^*$$

where c_j^{lcp} is the cost of least-cost path and is given as

$$c_j^{lcp} = \arg\min_{s \in \mathcal{S}} \min_{P \in \mathcal{P}_{js}} \left(\sum_{l \in P \cap \mathcal{L}} \gamma_l + 2e \cdot \sum_{k \in P \cap \mathcal{N}} \beta_k \right)$$

- Scheduling subproblem can be solved using linear programming.
- Let $\mathbf{p} = [\boldsymbol{\beta}, \boldsymbol{\gamma}]^T$ denote the price vector. Then, the price vector can be updated using the projected subgradient method as follows

$$\mathbf{p}[m+1] = [\mathbf{p}[m] - \delta \cdot \mathbf{g}(\mathbf{p}[m])]^+$$

Sufficiency of the clique constraints with respect to the optimal utility

Clique constraints are not sufficient to ensure conflict free schedules. However, we show that they are sufficient to optimally solve our initial resource allocation problem. As a consequence of this, we have the following propositions.

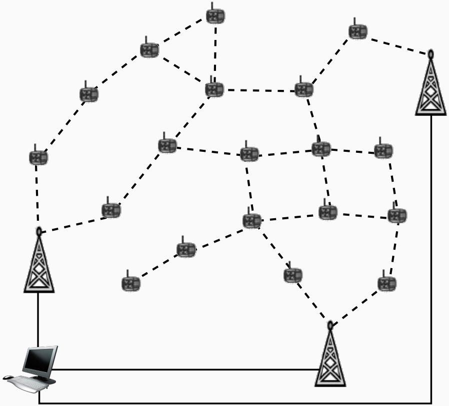
Proposition 1:

The optimal values of problems P_1 and P_2 are equal.

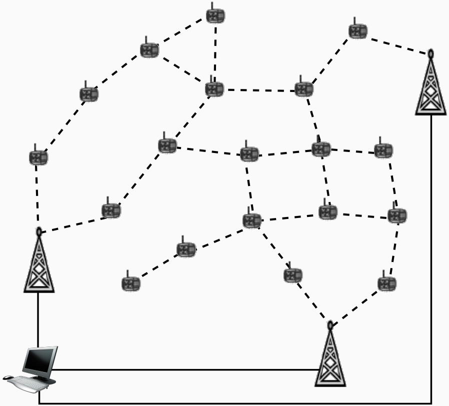
Proposition 2:

The projected subgradient method can be made to converge to an ϵ-band around the optimal solution of problem P_1.

Proof of Proposition 1 — an outline

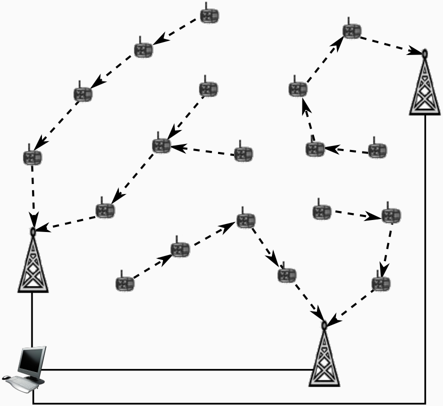


Proof of Proposition 1 — an outline



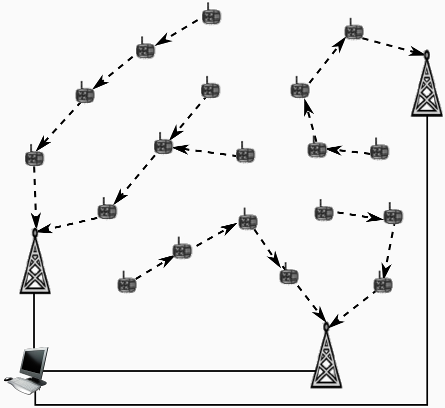
Let $\beta_j = 1 \, \forall j \in \mathcal{N}$ and $\gamma_l = 1 \, \forall l \in \mathcal{L}$

Proof of Proposition 1 — an outline



Let $\beta_j = 1\, \forall j \in \mathcal{N}$ and $\gamma_l = 1\, \forall l \in \mathcal{L}$

Proof of Proposition 1 — an outline



Observation: The collection of least cost paths forms a forest

Proof of Proposition 1 — an outline

- Since clique constraints are necessary but not sufficient, we have $P_1^{opt} \leq P_2^{opt}$.

Proof of Proposition 1 — an outline

- Since clique constraints are necessary but not sufficient, we have $P_1^{opt} \leq P_2^{opt}$.
- To show that $P_1^{opt} \geq P_2^{opt}$, consider an optimal price vector $[\beta^*, \gamma^*]$.

Proof of Proposition 1 — an outline

- Since clique constraints are necessary but not sufficient, we have $P_1^{opt} \leq P_2^{opt}$.
- To show that $P_1^{opt} \geq P_2^{opt}$, consider an optimal price vector $[\beta^*, \gamma^*]$.
- Consider the collection of least cost path from the nodes to the sink nodes.

Proof of Proposition 1 — an outline

- Since clique constraints are necessary but not sufficient, we have $P_1^{opt} \leq P_2^{opt}$.
- To show that $P_1^{opt} \geq P_2^{opt}$, consider an optimal price vector $[\beta^*, \gamma^*]$.
- Consider the collection of least cost path from the nodes to the sink nodes.
- Remove every other links in the network (these are not part of the optimal route).

Proof of Proposition 1 — an outline

- Since clique constraints are necessary but not sufficient, we have $P_1^{opt} \leq P_2^{opt}$.
- To show that $P_1^{opt} \geq P_2^{opt}$, consider an optimal price vector $[\beta^*, \gamma^*]$.
- Consider the collection of least cost path from the nodes to the sink nodes.
- Remove every other links in the network (these are not part of the optimal route).
- The resulting network is a forest - a perfect graph.

Proof of Proposition 1 — an outline

- Since clique constraints are necessary but not sufficient, we have $P_1^{opt} \leq P_2^{opt}$.
- To show that $P_1^{opt} \geq P_2^{opt}$, consider an optimal price vector $[\beta^*, \gamma^*]$.
- Consider the collection of least cost path from the nodes to the sink nodes.
- Remove every other links in the network (these are not part of the optimal route).
- The resulting network is a forest - a perfect graph.
- For a perfect graph, clique constraints are sufficient to ensure conflict-free schedules. Therefore, one can find valid schedules on the reduced network.

Proof of Proposition 1 — an outline

- Since clique constraints are necessary but not sufficient, we have $P_1^{opt} \leq P_2^{opt}$.
- To show that $P_1^{opt} \geq P_2^{opt}$, consider an optimal price vector $[\beta^*, \gamma^*]$.
- Consider the collection of least cost path from the nodes to the sink nodes.
- Remove every other links in the network (these are not part of the optimal route).
- The resulting network is a forest - a perfect graph.
- For a perfect graph, clique constraints are sufficient to ensure conflict-free schedules. Therefore, one can find valid schedules on the reduced network.
- The schedules in the reduced network remain valid in the original network.

Numerical evaluation

Numerical evaluation

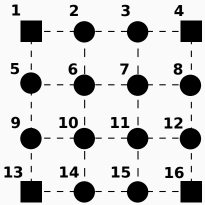


Figure 1: Network G_1

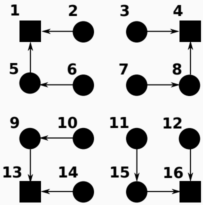
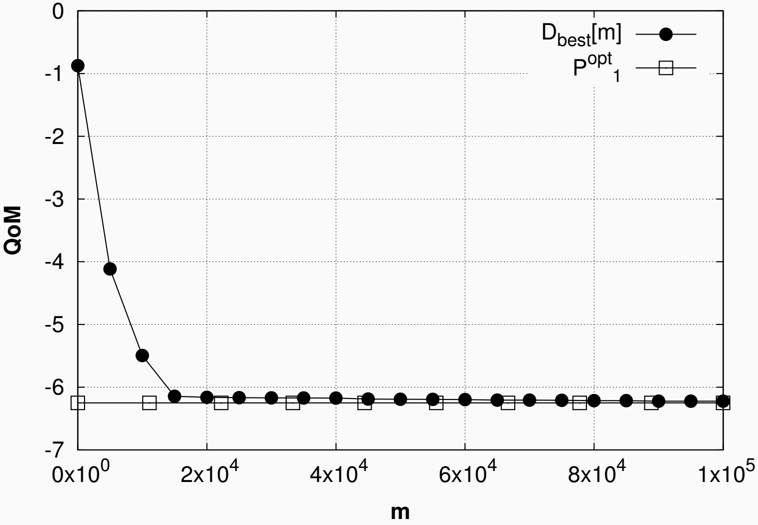


Figure 2: Optimal routes

Numerical evaluation



Numerical evaluation

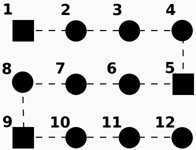


Figure 3: Network G_2

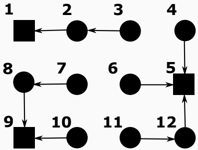
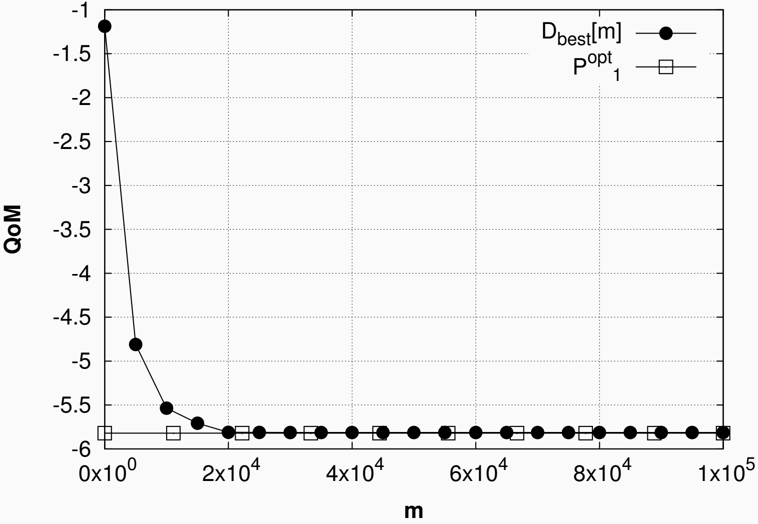


Figure 4: Optimal routes



-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
-5
-5.5
-6
QoM
0x10^0
2x10^4
4x10^4
6x10^4
8x10^4
1x10^5
m
D_best[m]
P^opt_1

Numerical evaluation

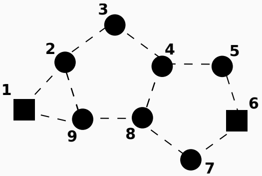


Figure 5: Network G_3

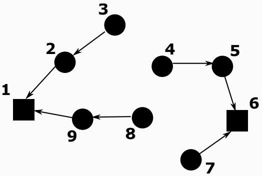
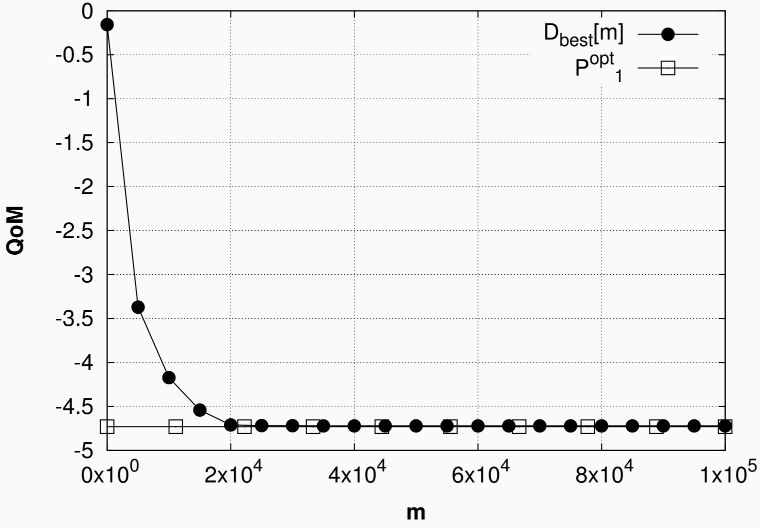


Figure 6: Optimal routes

Numerical evaluation




THANK
YOU