AdaSens: Adaptive Environment Monitoring by Coordinating Intermittently-Powered Sensors

Shuyue Lan , Zhilu Wang , John Mamish,
Josiah Hester, Qi Zhu

Speaker: Zhilu Wang
Northwestern University

Northwestern University

Problem

- How to adapt and coordinate a network of intermittently-powered sensors for better and more efficient environment perception.

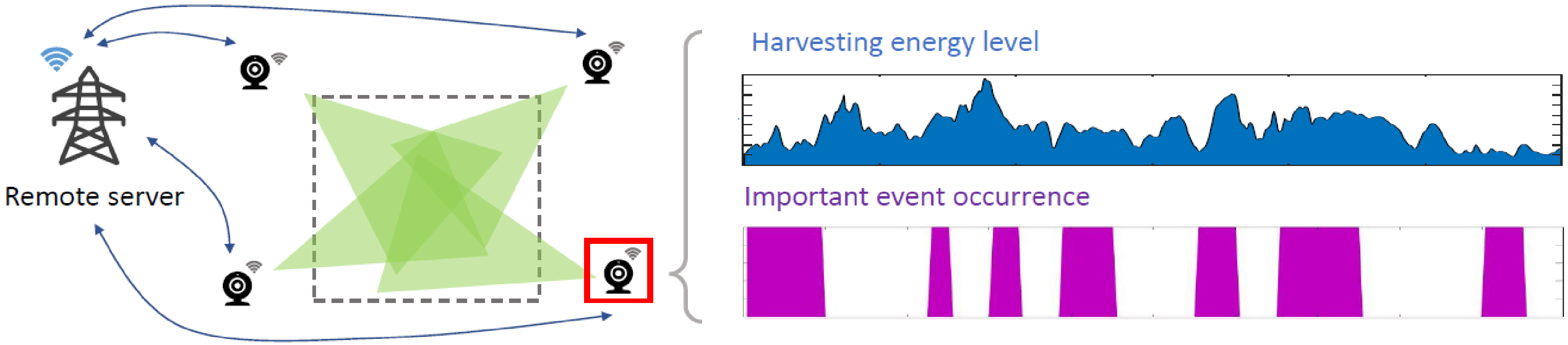


Fig. 1: The overview of environment monitoring on intermittently-powered sensor networks.

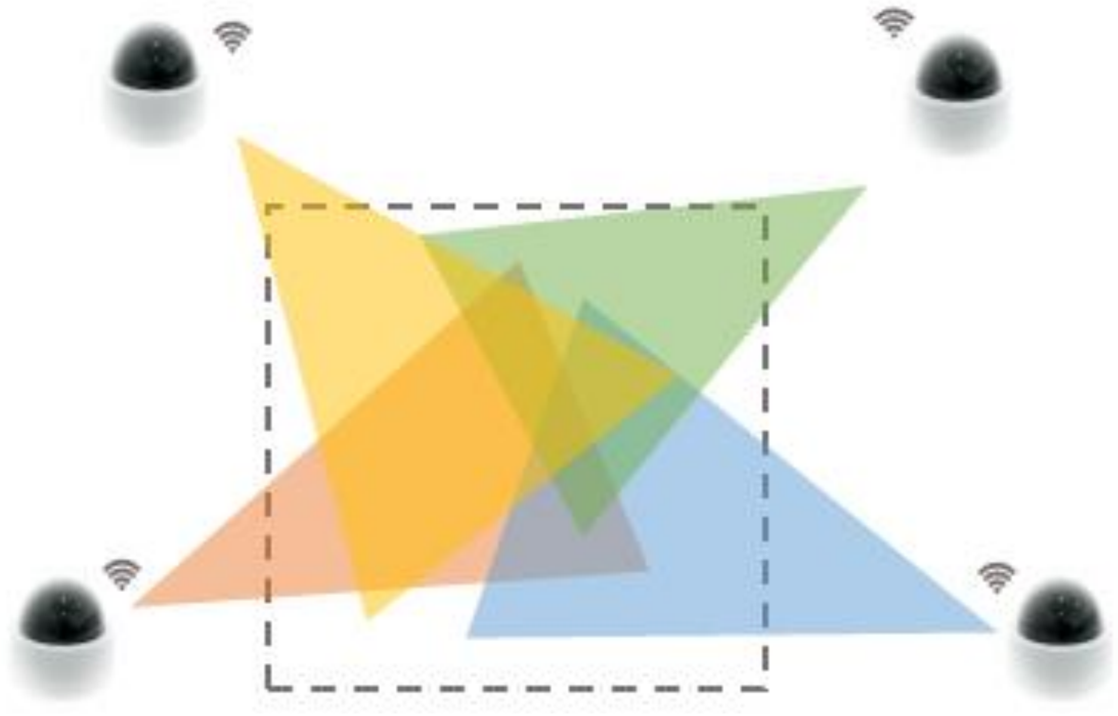
Challenges to Address

- How to leverage the redundancy among different agents (sensor nodes)
- How to work with limited resources on intermittently-powered devices for computation, communication and storage
- How to overcome the unstable energy supply
- How to coordinate and utilize all nodes to better cover the important scenes

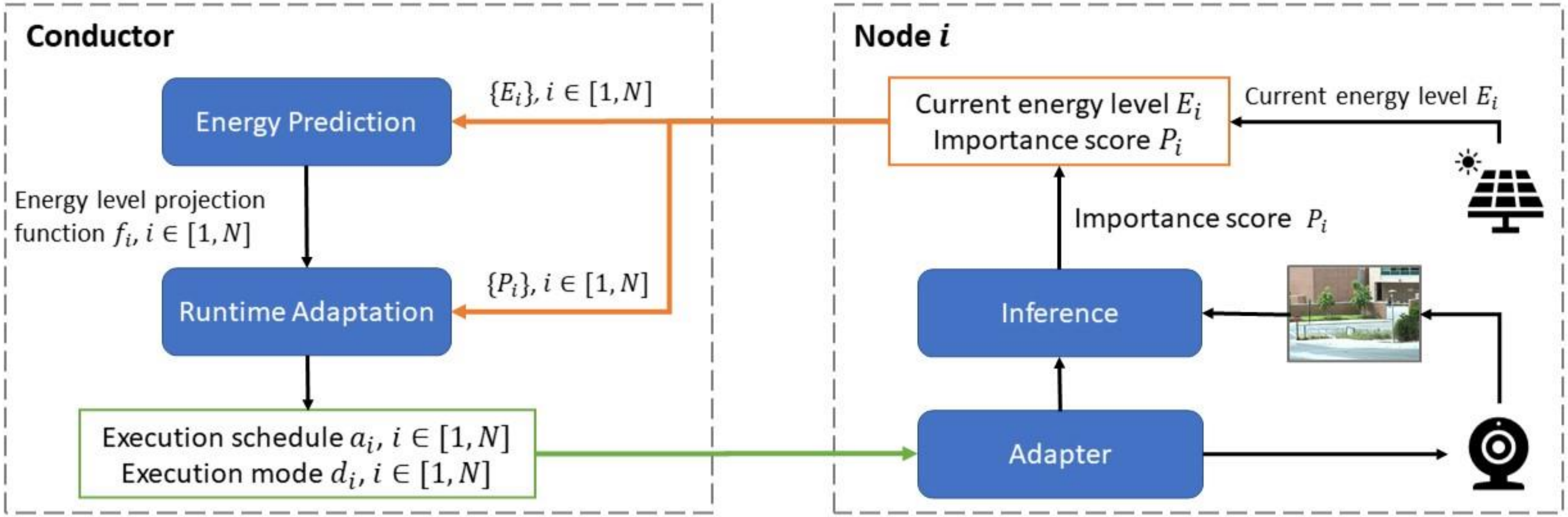
AdaSens

- Target:
  - An environment monitoring system that covers as much as possible of the targeted scene with intermittently-powered sensors.

- Setup:
  - Multiple low-power sensor nodes
    - Cameras/other sensors
    - Energy harvesting module
  - One conductor node
    - Enough resources
    - Communicate with sensor nodes
    - Coordinate node executions



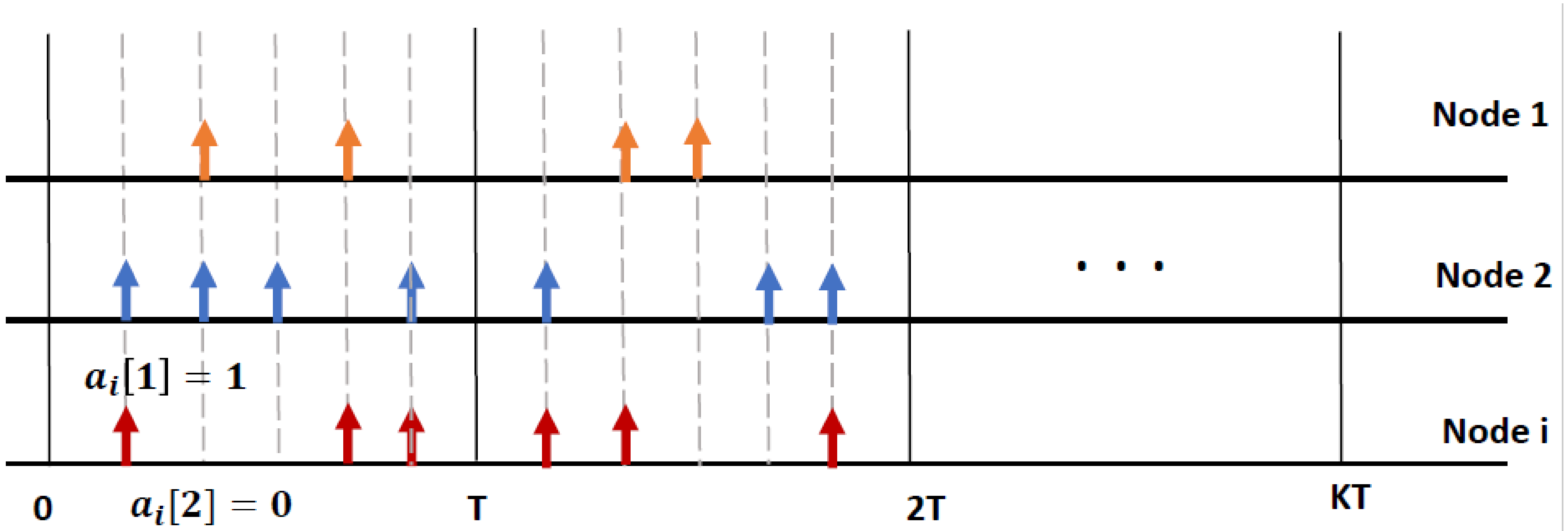
Overview of AdaSens Framework



Conductor: schedule the operations of individual sensor nodes.

Node: receive the schedule of executions from the conductor and executes accordingly.

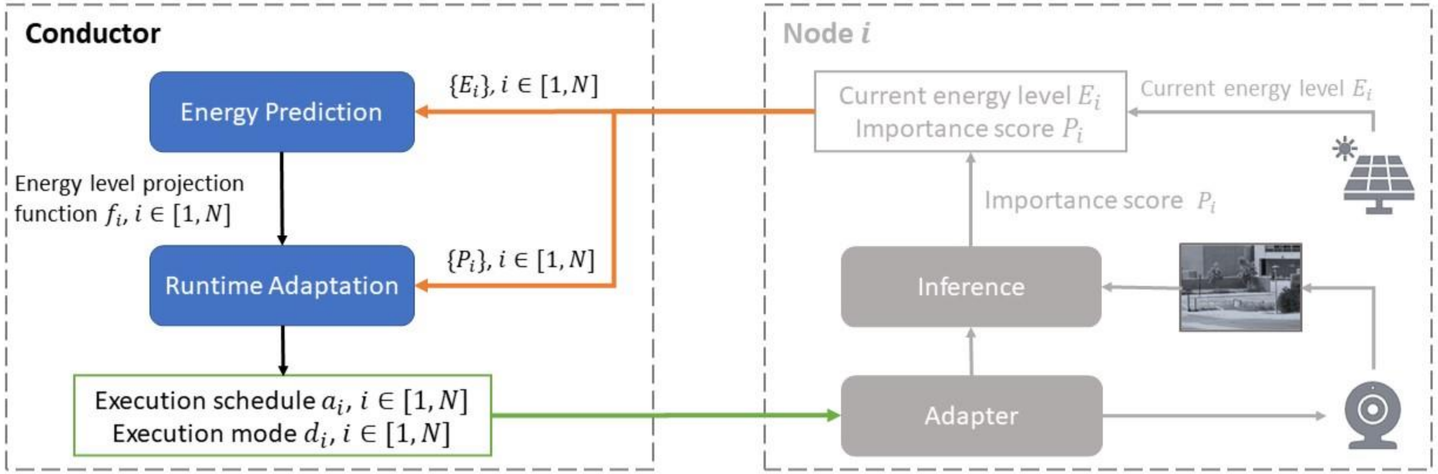
Adaptation Variables



At the beginning of each adaptation period, the conductor decides whether to do an atomic task at time t for each node i, indicated as $a_i[t]$.
$a_i[t] = 1$: node i is instructed to execute one task at the time step t.

Conductor

- Task:
  - Schedule the operations of individual sensor nodes, with the consideration of the scene coverage.
- Components:
  - Energy prediction
  - Runtime adaptation



Conductor – Energy Prediction

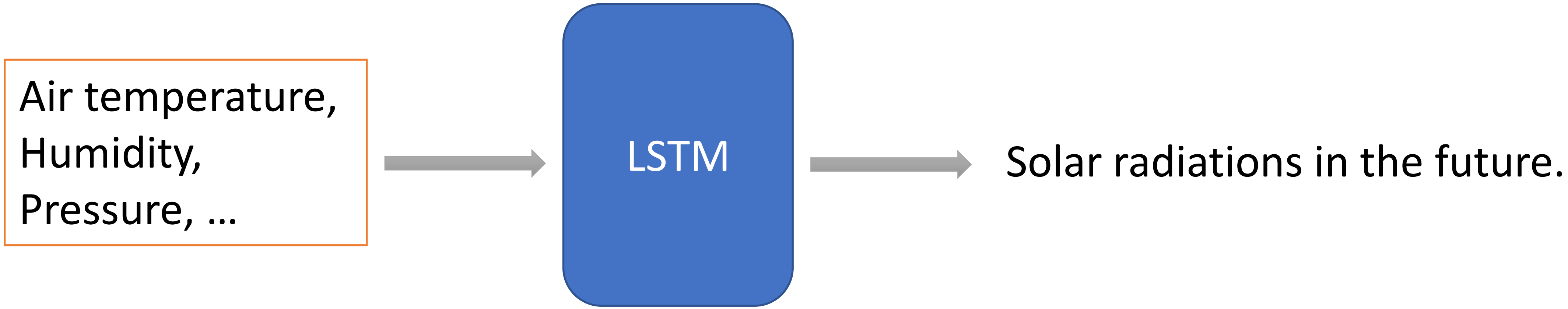
- Input:
  - Environment information.
  - The current energy level E_i of sensor node i.
- Energy level projection function at time t for node i

$$f_i\left(t, E_i^{used}\right) = E_i + \sum_{j=0}^{t-1} \Delta ec_i[j] - E_i^{used}$$

  - $\Delta ec_i[j]$: the energy creation rate of sensor node i at time j.

Conductor – Energy Prediction

- For solar power, $\Delta ec_i[j] = P_{solar}[j] = AS[j]$
  - $S[j]$ is the solar radiation at time j, which is predicted by Long Short-Term Memory model (LSTM).



Conductor – Runtime Adaptation

$$\max_{\substack{\forall a_i[t],\\ i\in[1,N],\\ t\in[0,KT)}} \sum_{t=0}^{KT-1} \sum_{m=1}^{M} x_m \cdot \max_{i\in[1,N]} (\lambda_{i,m} \cdot p_i[t] \cdot a_i[t])$$

$$\text{s.t. for } i \in [1, N],\ t \in [0, KT)$$

$$f_i[t] = E_i + \sum_{j=0}^{t-1} (\Delta ec_i[j] - \Delta eu_i \cdot a_i[j]) - \left\lfloor \frac{t}{T} \right\rfloor E_c,$$

$$f_i[t] \le C$$

$$f_i[t] \ge \Delta eu_i \times a_i[t], \quad t \ne kT - 1,\ k \in [1, K]$$

$$f_i[t] \ge \Delta eu_i \times a_i[t] + E_{comm}, \quad t = kT - 1$$

- Objective
  - Maximize the spatial-temporal coverage of the targeted scene.
    - x_m: area of the m_{th} scene partition.
    - $\lambda_{i,m}$: indicate whether the area x_m is in the view of node i.
    - $p_i[t]$: the decay importance score of node i at time t.

$$p_i[t] = 0.5 + (P_i - 0.5) \cdot \exp(-\frac{t}{\rho})$$

Conductor – Runtime Adaptation

$$\max_{\substack{\forall a_i[t],\\ i\in[1,N],\\ t\in[0,KT)}} \sum_{t=0}^{KT-1} \sum_{m=1}^{M} x_m \cdot \max_{i\in[1,N]} (\lambda_{i,m} \cdot p_i[t] \cdot a_i[t])$$

$$\text{s.t. for } i \in [1,N],\ t \in [0,KT)$$

$$f_i[t] = E_i + \sum_{j=0}^{t-1} (\Delta ec_i[j] - \Delta eu_i \cdot a_i[j]) - \left\lfloor \frac{t}{T} \right\rfloor E_c,$$

$$f_i[t] \le C$$

$$f_i[t] \ge \Delta eu_i \times a_i[t], \quad t \ne kT-1,\ k \in [1,K]$$

$$f_i[t] \ge \Delta eu_i \times a_i[t] + E_{comm}, \quad t = kT-1$$

- Energy dynamic function
  - Compute the projected energy level of node i at time t.

$$E_i^{used}$$

$$\downarrow$$

$$\sum_{j=0}^{t-1} \Delta eu_i \cdot a_i[j] + \left\lfloor \frac{t}{T} \right\rfloor E_c$$

Conductor – Runtime Adaptation

$$\max_{\substack{\forall a_i[t],\\ i\in[1,N],\\ t\in[0,KT)}} \sum_{t=0}^{KT-1} \sum_{m=1}^{M} x_m \cdot \max_{i\in[1,N]} (\lambda_{i,m} \cdot p_i[t] \cdot a_i[t])$$

$$\text{s.t. for } i \in [1, N],\ t \in [0, KT)$$

$$f_i[t] = E_i + \sum_{j=0}^{t-1} (\Delta ec_i[j] - \Delta eu_i \cdot a_i[j]) - \left\lfloor \frac{t}{T} \right\rfloor E_c,$$

$$f_i[t] \leq C$$

$$f_i[t] \geq \Delta eu_i \times a_i[t], \quad t \neq kT - 1,\ k \in [1, K]$$

$$f_i[t] \geq \Delta eu_i \times a_i[t] + E_{comm}, \quad t = kT - 1$$

- Energy capacity constraint
  - The predicted available energy of each node at every time step is within the designed energy capacity.

Conductor – Runtime Adaptation

$$\max_{\substack{\forall a_i[t],\\ i\in[1,N],\\ t\in[0,KT)}} \sum_{t=0}^{KT-1} \sum_{m=1}^{M} x_m \cdot \max_{i\in[1,N]} (\lambda_{i,m} \cdot p_i[t] \cdot a_i[t])$$

$$\text{s.t. for } i \in [1,N],\ t \in [0,KT)$$

$$f_i[t] = E_i + \sum_{j=0}^{t-1} (\Delta ec_i[j] - \Delta eu_i \cdot a_i[j]) - \left\lfloor \frac{t}{T} \right\rfloor E_c,$$

$$f_i[t] \leq C$$

$$\boxed{f_i[t] \geq \Delta eu_i \times a_i[t], \quad t \neq kT-1,\ k \in [1,K]}$$

$$f_i[t] \geq \Delta eu_i \times a_i[t] + E_{comm}, \quad t = kT-1$$

- Execution constraint
  - For each time step, the available energy should be no less than the energy consumption of executing a task.

Conductor – Runtime Adaptation

$$\max_{\substack{\forall a_i[t],\\ i\in[1,N],\\ t\in[0,KT)}} \sum_{t=0}^{KT-1} \sum_{m=1}^{M} x_m \cdot \max_{i\in[1,N]} (\lambda_{i,m} \cdot p_i[t] \cdot a_i[t])$$

$$\text{s.t. for } i \in [1,N],\ t \in [0,KT)$$

$$f_i[t] = E_i + \sum_{j=0}^{t-1} (\Delta ec_i[j] - \Delta eu_i \cdot a_i[j]) - \left\lfloor \frac{t}{T} \right\rfloor E_c,$$

$$f_i[t] \le C$$

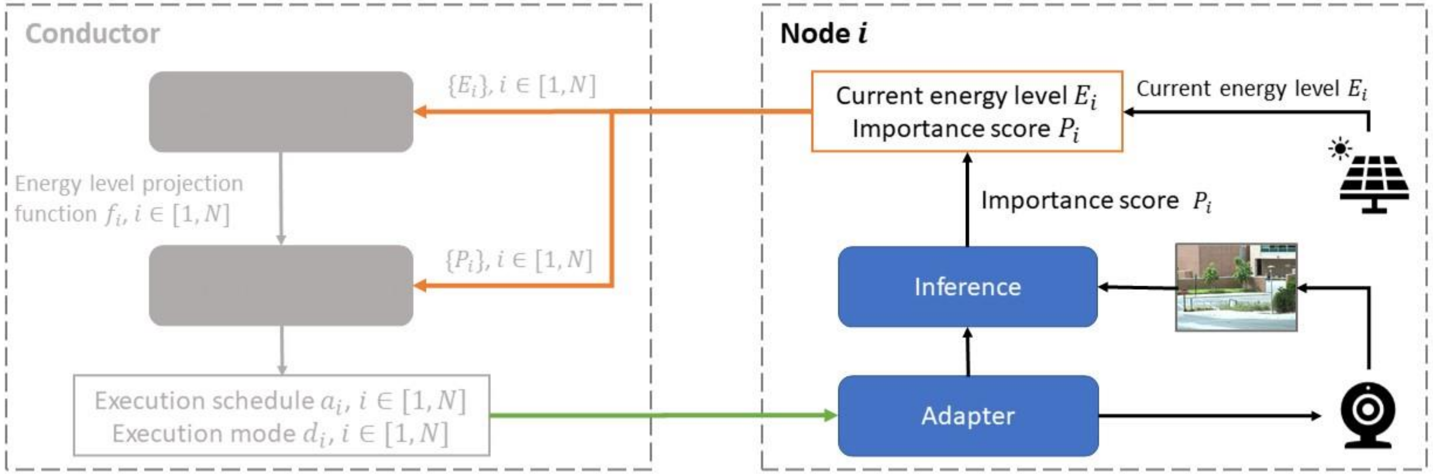
$$f_i[t] \ge \Delta eu_i \times a_i[t], \quad t \ne kT-1,\ k \in [1,K]$$

$$f_i[t] \ge \Delta eu_i \times a_i[t] + E_{comm}, \quad t = kT-1$$

- Execution constraint with communication
  - At the last time step in a period, the available energy should be no less than energy cost of scheduled executions and one-time communication cost.

Sensor Node

- Task:
  - Receive the schedule of executions from the conductor and executes accordingly.
- Components:
  - Adapter
  - Inference



Node – Adapter

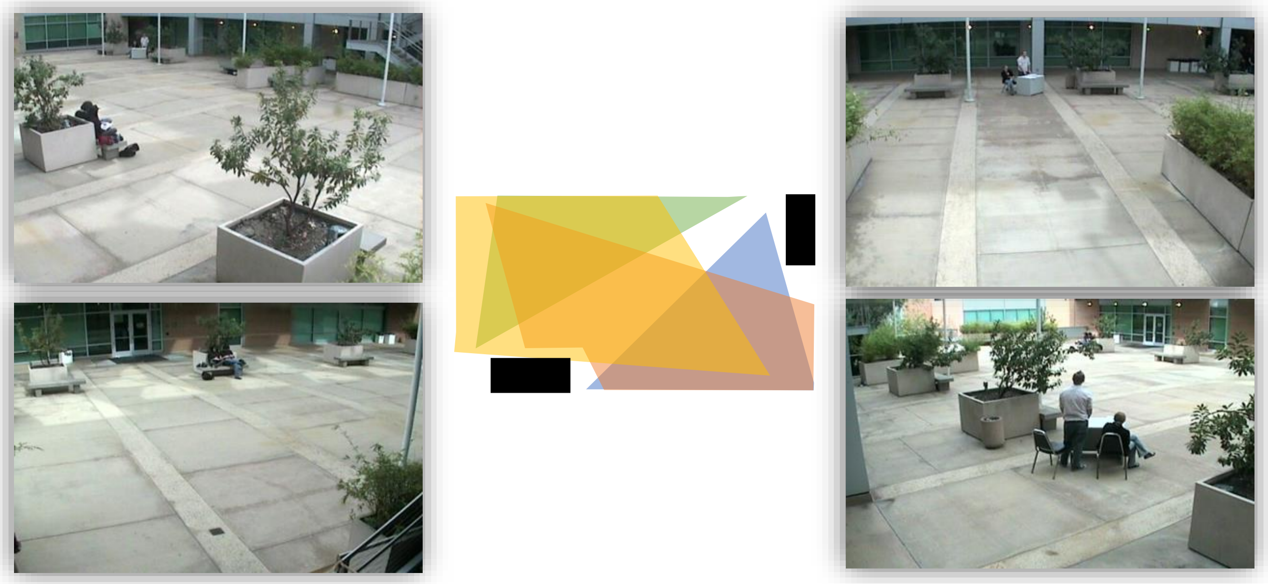
- In-network mode
  - A node has enough energy to communicate with the conductor and update its status.
  - The node will execute according to the instructions from the conductor.
- Isolation mode
  - A node does not have enough energy to communicate with the conductor and the conductor does not know the status of it.
  - The node will execute as many tasks as it can based on its own energy availability.
    - Accumulate the one-time communication energy before starting to execute greedily.

Node – Inference

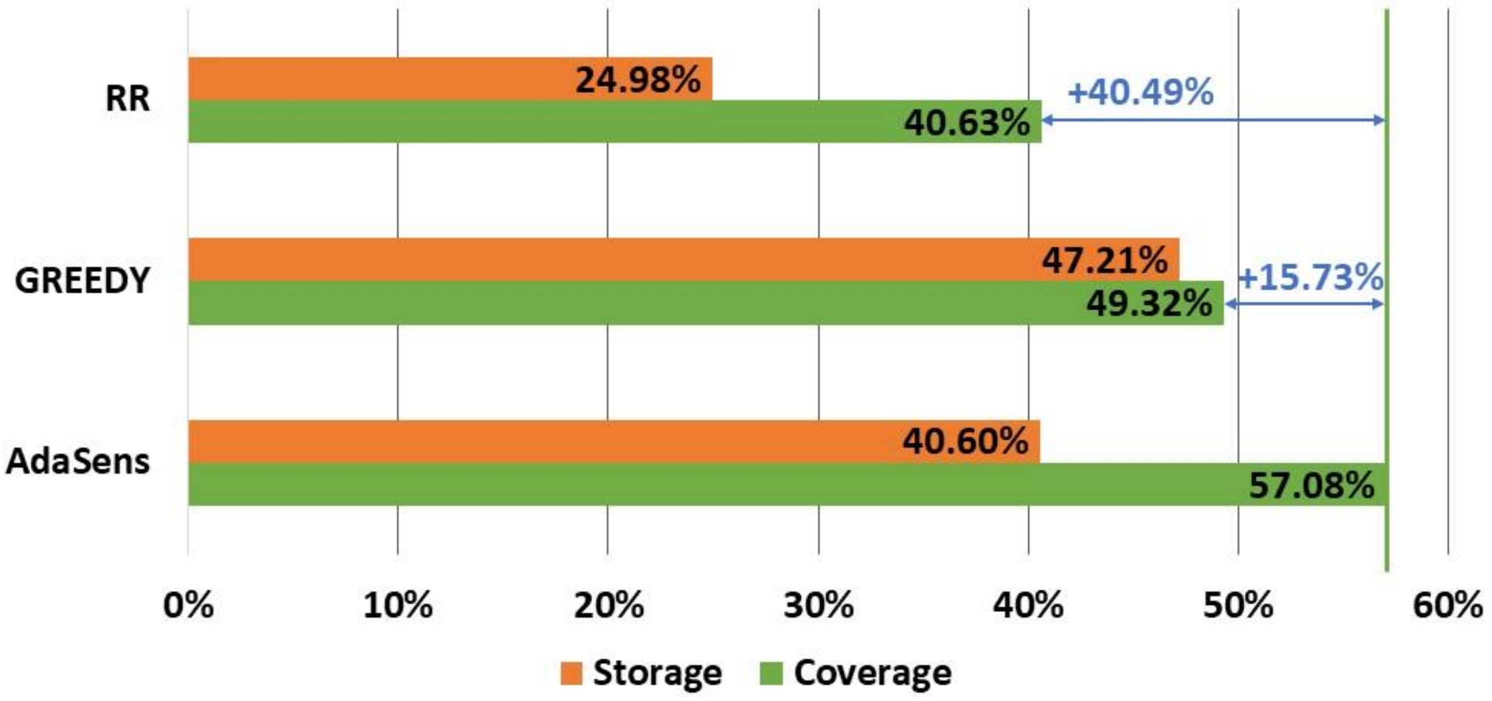
- Input
  - The captured images from the camera.
- Output
  - The importance of frames, given as a probability $P_i[t]$.
  - The importance of the frames is defined according to the application.

Experiments

- Datasets:
  - VideoWeb: Day 1 subset.
    - 8 different scenes, and 6 views of videos.
- Performance Measure:
  - Coverage in frame level.
  - The storage.

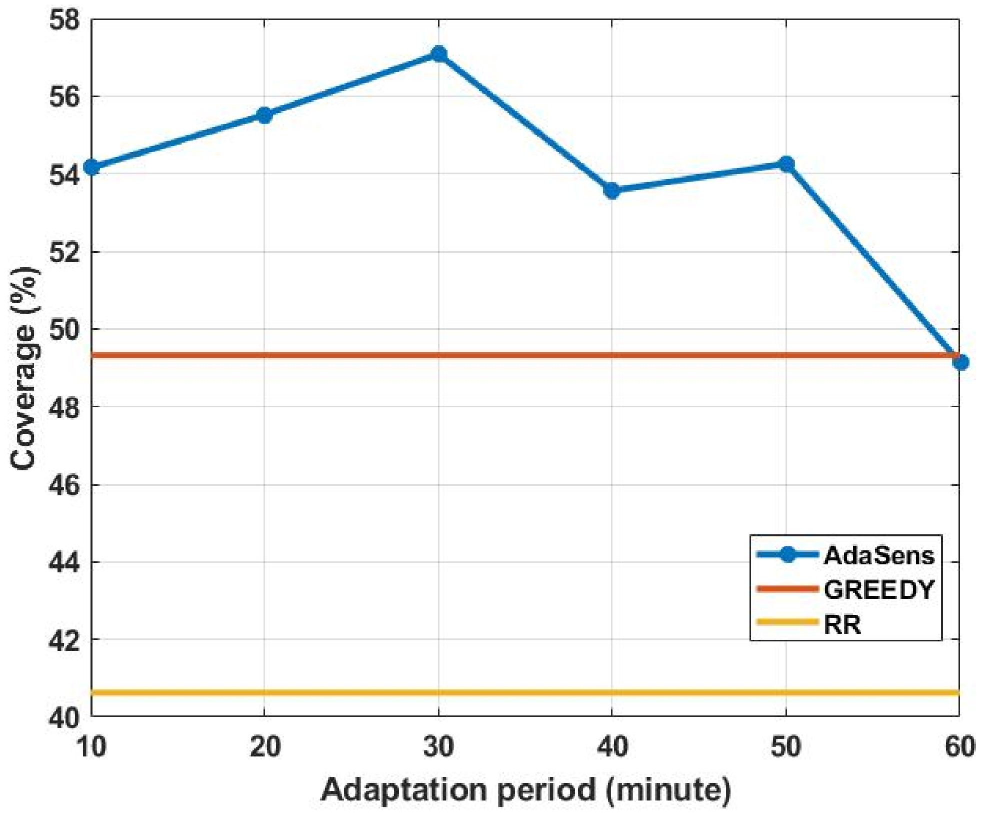


Result – Coverage and Storage on VideoWeb



- On average, AdaSens achieves a coverage of 57.08% of important frames with 40.60% of processed and stored frames.

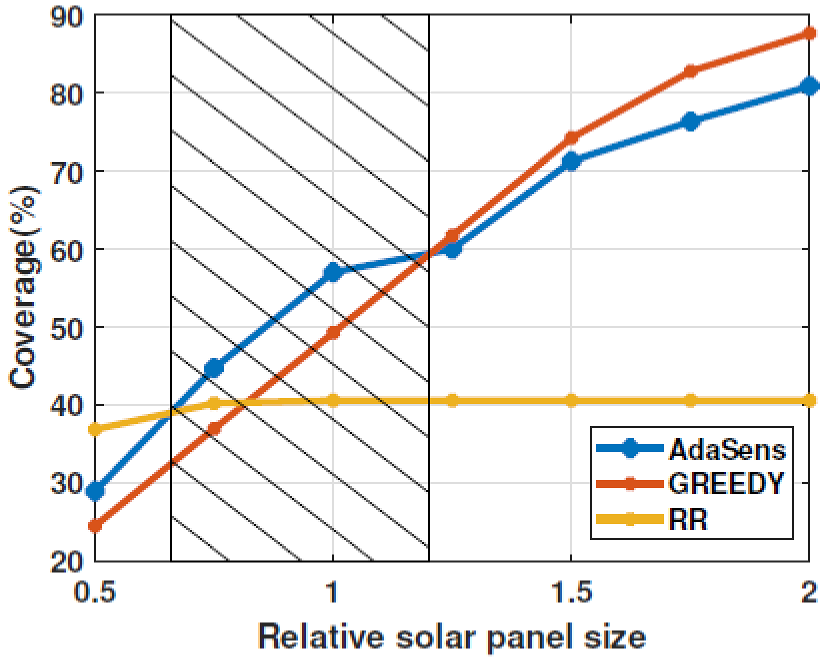
Result – Impact of T and K



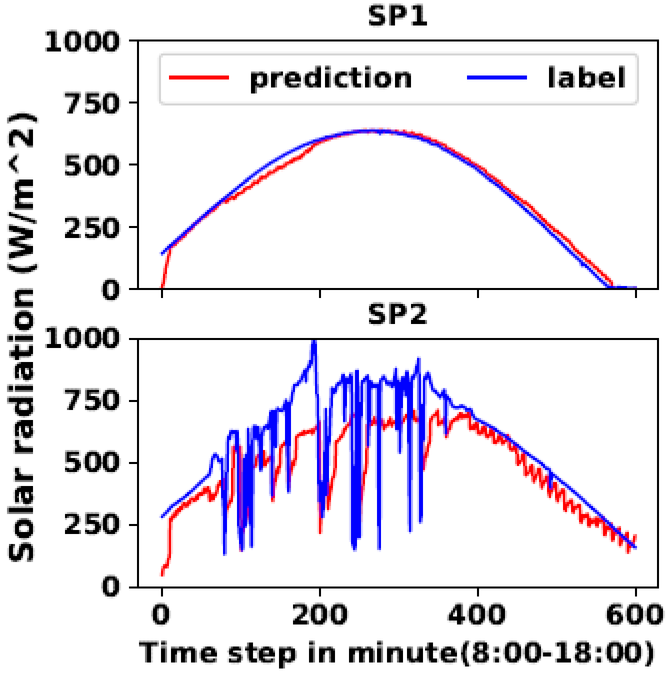
- At first, with the increase of adaptation period T, the coverage increases.
  - Less energy for communication and more energy for computation.
  - Longer optimization horizon.
- The coverage decreases when T keeps growing.
  - Less accurate solar energy prediction.
  - Out-of-date important scores of views.

Similar trend is observed for K.

Result – Impact of Energy Related Factors



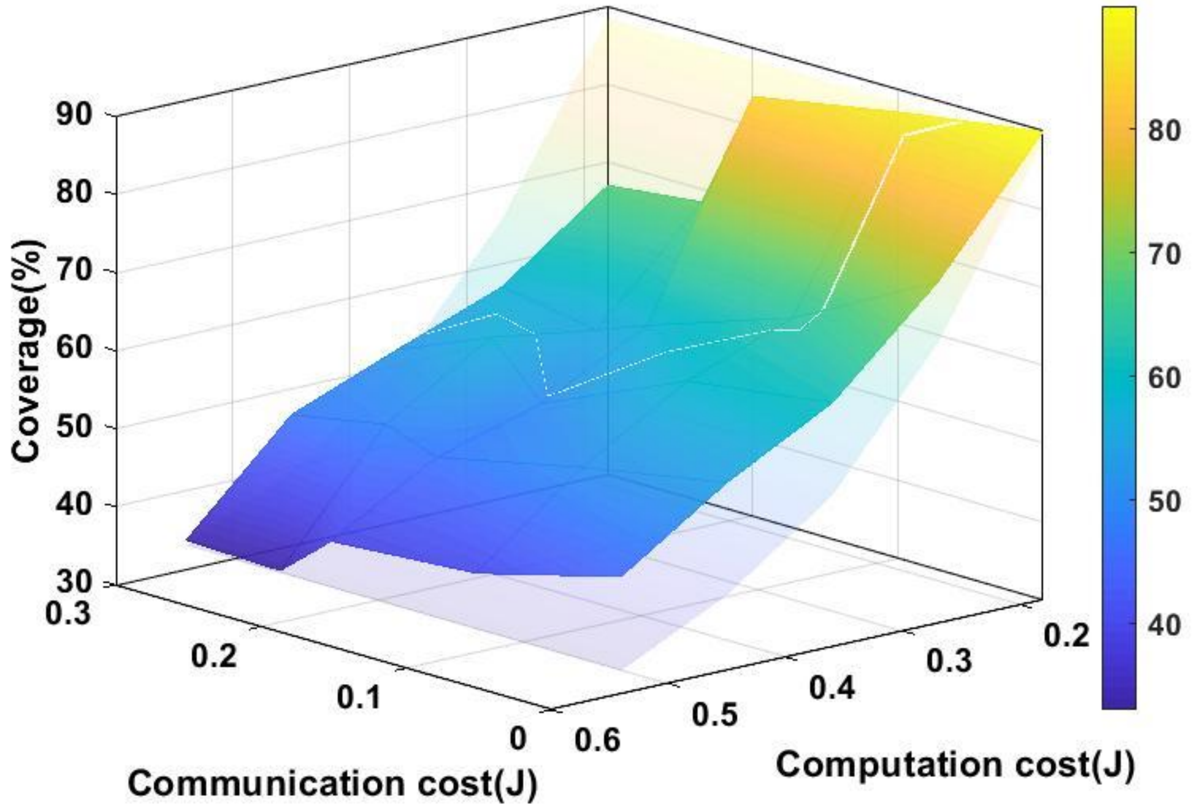
(a) Impact of energy availability.



(b) Examples of solar profiles.

Profile	GREEDY	AdaSens	Coverage gain
SP1	48.36%	58.33%	+20.62%
SP2	49.32%	57.08%	+15.73%

Result – Different Communication and Computation Costs



The ratio of communication cost versus computation cost is smaller.

↓

More benefits on coverage.

Thanks.