Information Freshness for Monitoring and Control Over Wireless Networks

Vishrant Tripathi

8th August, 2023

Outline

I. Introduction

1.What is Age of Information (AoI)?

2.How does it relate to monitoring and control?

II. Functions of AoI

How to optimize general functions of AoI in single-hop and multi-hop wireless networks?

- Whittle AoI – Allerton, 2019
- Age Debt – Infocom, 2021

III. Online Learning

What happens when the functions of AoI are unknown, time-varying and possibly adversarial?

- Online Learning – MobiHoc, 2021

IV. Applications

1.Multi-Agent occupancy grid mapping - simulations

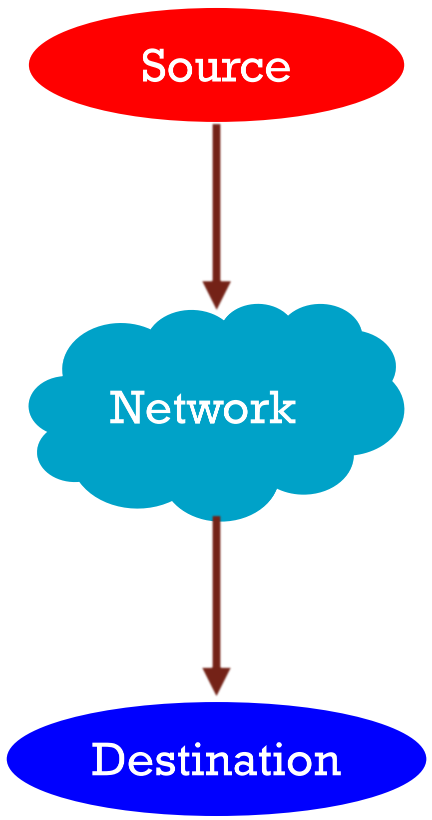
2.Multi-UAV mobility tracking - experiments

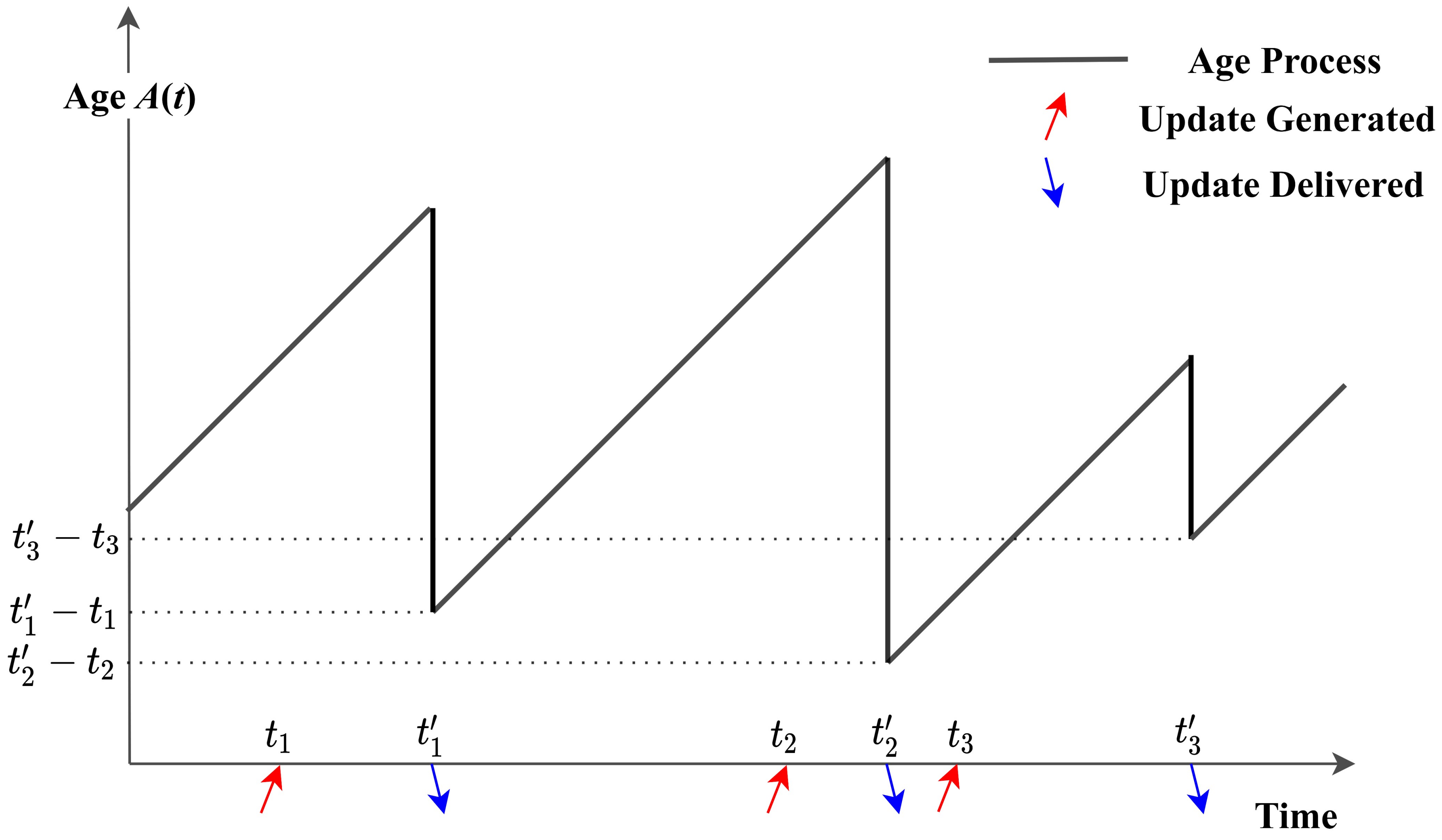
- Tradeoffs – WiOpt, 2021
- WiSwarm – Infocom, 2023

PART I

INTRODUCTION

Age of Information (AoI)



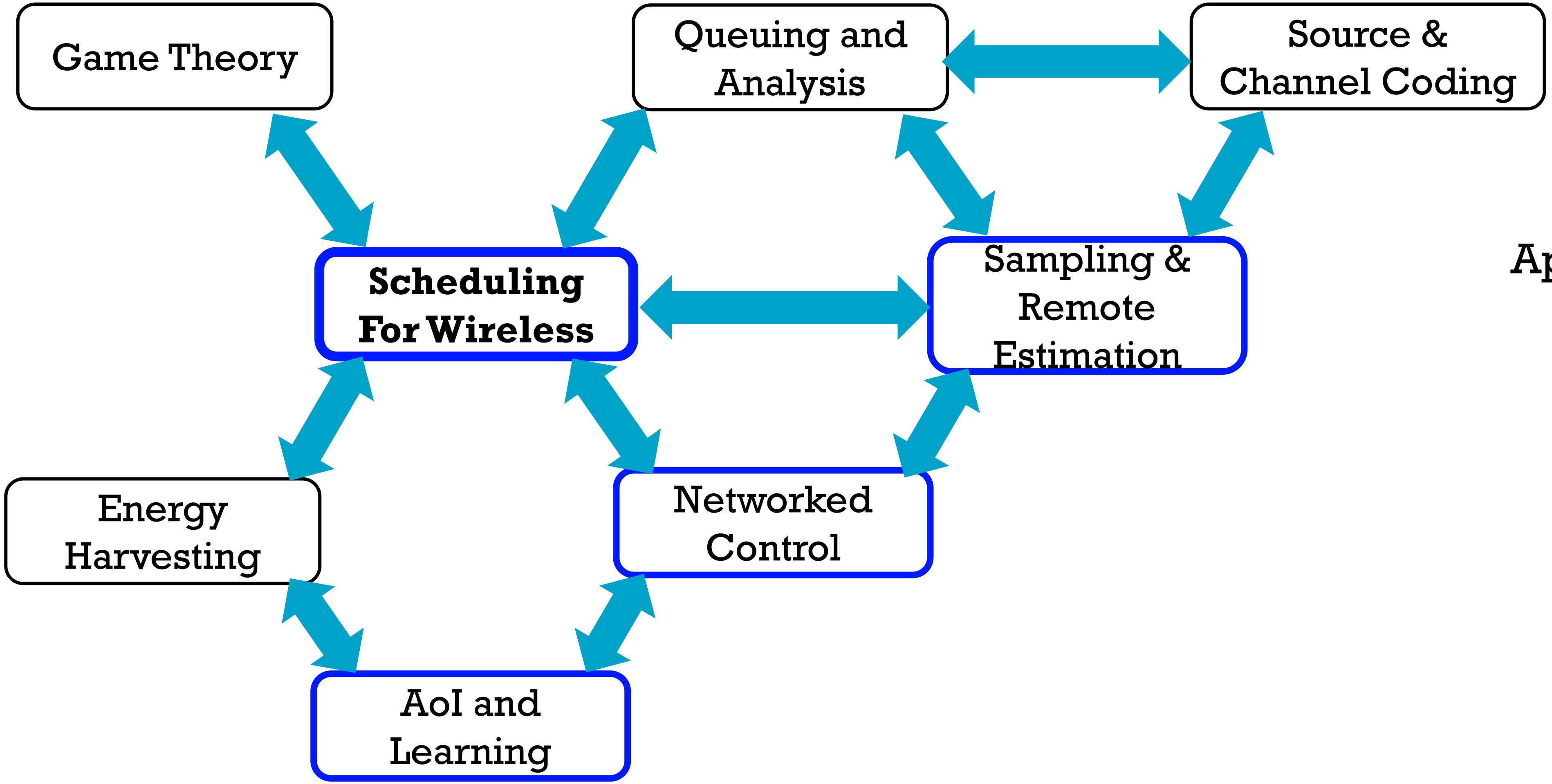


Motivation: Why Study AoI?

- Real-time applications
  - sensing for IoT applications
  - control of robot swarms
  - vehicle-to-vehicle (V2V) communication
  - surveillance and monitoring
- Having the freshest available data is essential to system performance
- AoI formalizes the notion of freshness
- Low AoI ⇒ fresh information ⇒ better performance

A Quick Tour

Of The AoI Map



Applications

- Federated learning
- **Robotics**
- **Edge Computing**
- IoT
- Caching
- …

http://webhome.auburn.edu/~yzs0078/AoI.html

PART II

FUNCTIONS OF AGE

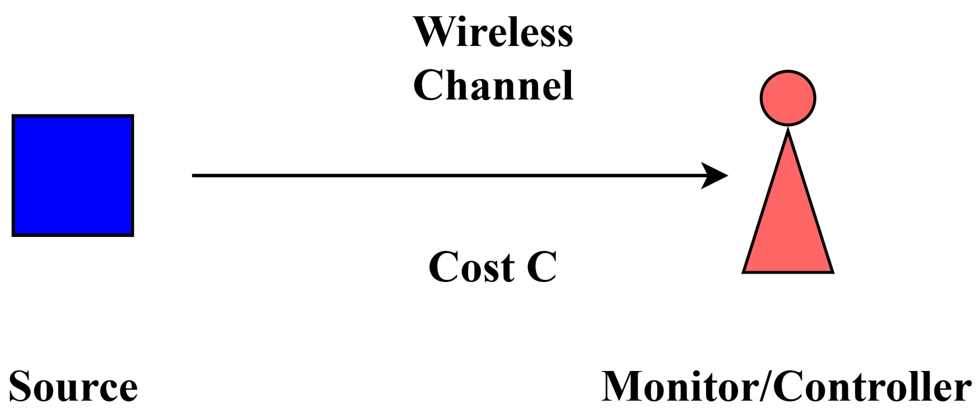
Warmup: Monitoring A Linear System

- LTI system

$$x_{t+1} = Ax_t + w_t, \text{ where } w_t \sim \mathcal{N}(0, \Sigma)$$

- Observing the source has a cost C
- Monitor maintains estimates $\hat{\boldsymbol{x}}_t$
- **Goal**: Minimize monitoring error + sampling cost, i.e.

$$\min_{u_1,\dots,u_T} \sum_{t=1}^{T} \left(||x_t - \hat{x}_t||^2 + Cu_t \right)$$



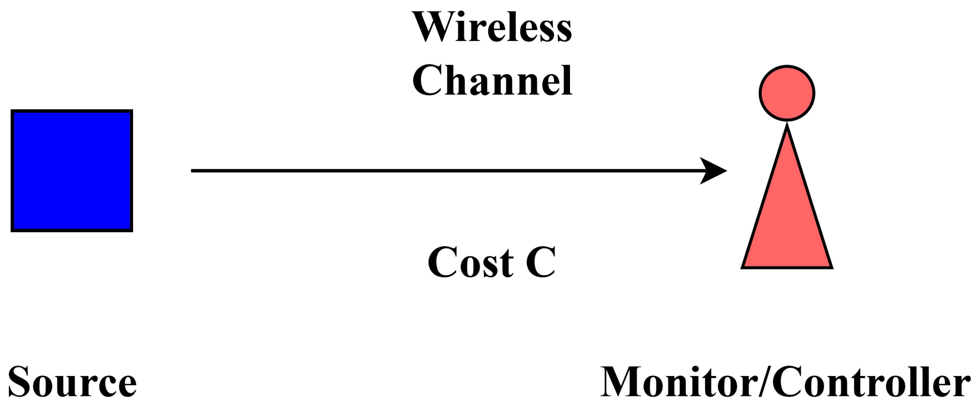
Warmup: Monitoring A Linear System

- **Key Observation**: If last update was received τ time-slots ago, then

$$\mathbb{E}\left[||x_t - \hat{x}_t||^2\right] = f(\tau) = \sum_{k=0}^{\tau-1} Tr(A^{k^\intercal} A^k \Sigma)$$

- Monitoring error only depends on how long it has been since last update was received, i.e. **AoI**

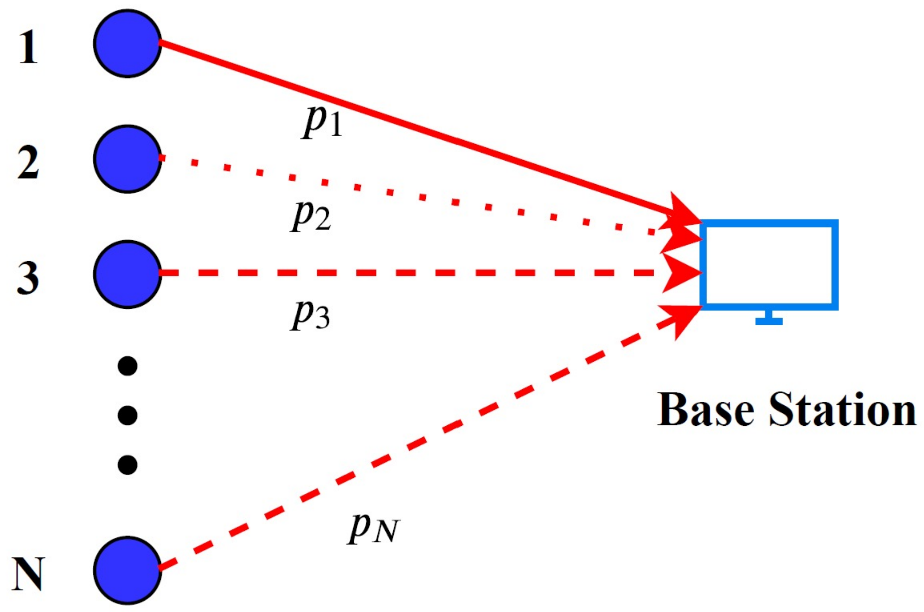
- **Minimizing error ⇔ Minimizing a function of AoI**



Our Model

- N sources sending updates to a base station, only one can transmit at a time
- $A_i(t)$ measures how long it has been since the base station received an update about source i
- Cost function $f_i(A_i(t))$ maps AoI to monitoring or control performance
- **Assumption:** $f_i(\cdot)$ is monotonically increasing
- **Goal:** find a scheduling policy that solves

$$\min_{\pi}\left(\lim_{T\to\infty}\frac{1}{T}\mathbb{E}\left[\sum_{i=1}^{N}\sum_{t=1}^{T} f_i(A_i^{\pi}(t))\right]\right)$$



The Whittle Index Approach

Whittle Index - low complexity heuristic with good performance for Restless Multi-Armed Bandit (RMAB) problems.

Steps

1. Formulate scheduling/allocation problem as a RMAB
2. Use a Lagrangian relaxation to formulate the decoupled problem (*dcp*)
3. Solve the *dcp* and establish *indexability*
4. Find an expression for the Whittle Index

Indexability and the Whittle Index

- **Whittle (1980s)** – near-optimal policy can be computed **efficiently**, given special **indexability** property

- **Indexability of Arm i:** Given activation cost $C > 0$, the set of states for which it is optimal to activate the arm decreases **monotonically** as C increases

- Compute index functions $W_i \colon S_i \to \mathbb{R}$, which denote how "valuable" it is to activate arm i at state s_i

Whittle Index for AoI

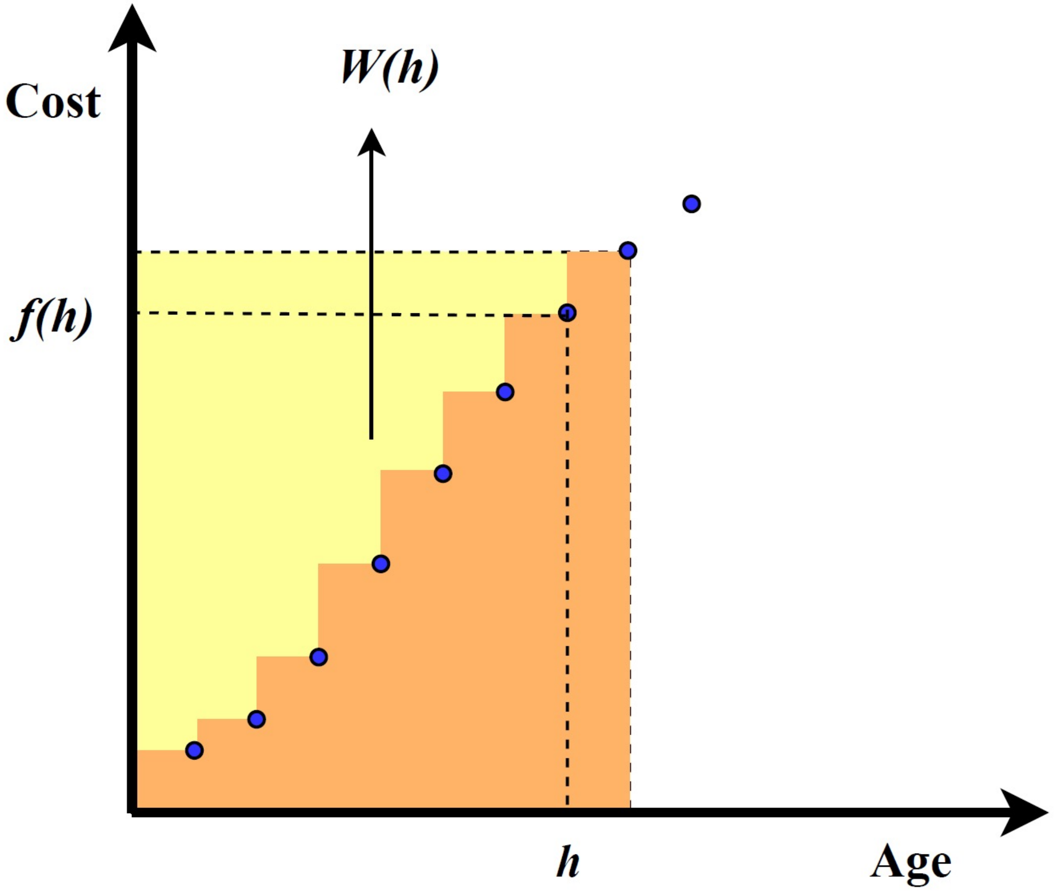
- We establish that the functions of AoI problem is *indexable*

- Whittle Index expression for source i (given reliable channels)

$$W_i(h) = h\, f_i(h+1) - \sum_{j=1}^{h} f_i(j)$$

- **Whittle Index policy**

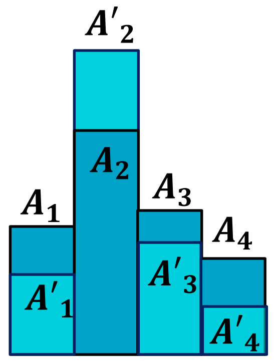
$$\pi(t) = \arg\max_i \{W_i(A_i(t))\}$$



Performance Guarantees

- Typically, Whittle Index is **not optimal**
- Optimality results available for
  - symmetric systems using a coupling argument
  - In the limit as system sizes go to infinity via a fluid limit approach
- We establish optimality of the Whittle index for a **finite asymmetric system** **(N = 2)**
- We also show that **some** index policy is optimal for **N=3**
- Through simulations we observe
  - Whittle is not exactly, but very close to optimal for **N=4**
  - Whittle is **close to optimal** for large system sizes

A Recipe for Whittle Optimality



- Define **strong-switch-type policies**

$$\forall \vec{A} \text{ and } \vec{A'} \text{ such that } \boldsymbol{A_i \geq A'_i} \text{ and } \boldsymbol{A_j \leq A'_j \; \forall j \neq i}$$
$$\text{If } \boldsymbol{\pi(A_1, \ldots, A_N) = i}, \text{ then } \boldsymbol{\pi(A'_1, \ldots, A'_N) = i}$$

- Show that there exists an optimal policy of strong-switch-type

- Define **index policies** - for any set of monotone functions $F_1, \ldots F_N$

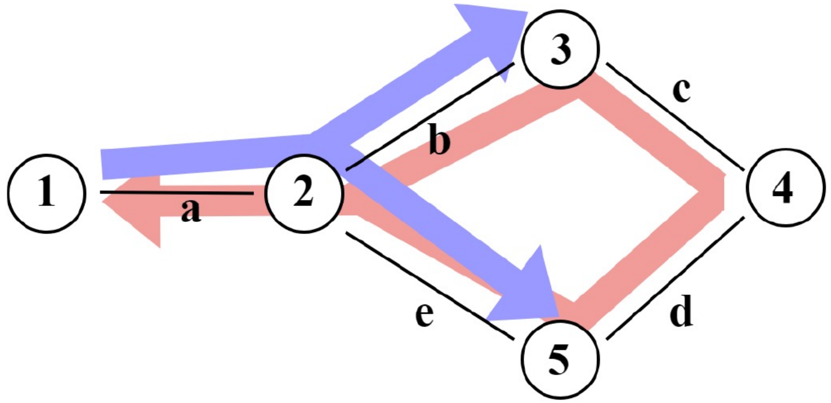
$$\boldsymbol{\pi(A_1, \ldots A_N) = \arg\max_i \{ F_i(A_i) \}}$$

- Show that index policies are **equivalent** to strong-switch-type policies

- Show that Whittle Index is the best among index policies **(N=2)**

The Multi-Hop Problem

- General wireless network with unicast, multicast and broadcast flows, and functions of AoI
- Need to optimize both scheduling and routing decisions
- **Age Debt** – A modified Lyapunov drift approach, similar to Proportional Integral control around a set point
- **Heuristic**, but best performing policy for all single-hop and multi-hop settings studied in literature until now



Age Debt: A Quick Primer

- Network admin provides **target average AoI** cost value α_i for each flow i

- Set up virtual queues (debt queues) of the form

$$Q_i(t+1) = \left[Q_i(t) + f_i\big(A_i(t+1)\big) - \alpha_i \right]^+$$

- Set up the Lyapunov function $L(t) = \sum_{i=1}^{N} Q_i^2(t)$

- **Age Debt Policy (*best known policy for multi-hop*) -** minimize the Lyapunov drift

PART III

ONLINE LEARNING

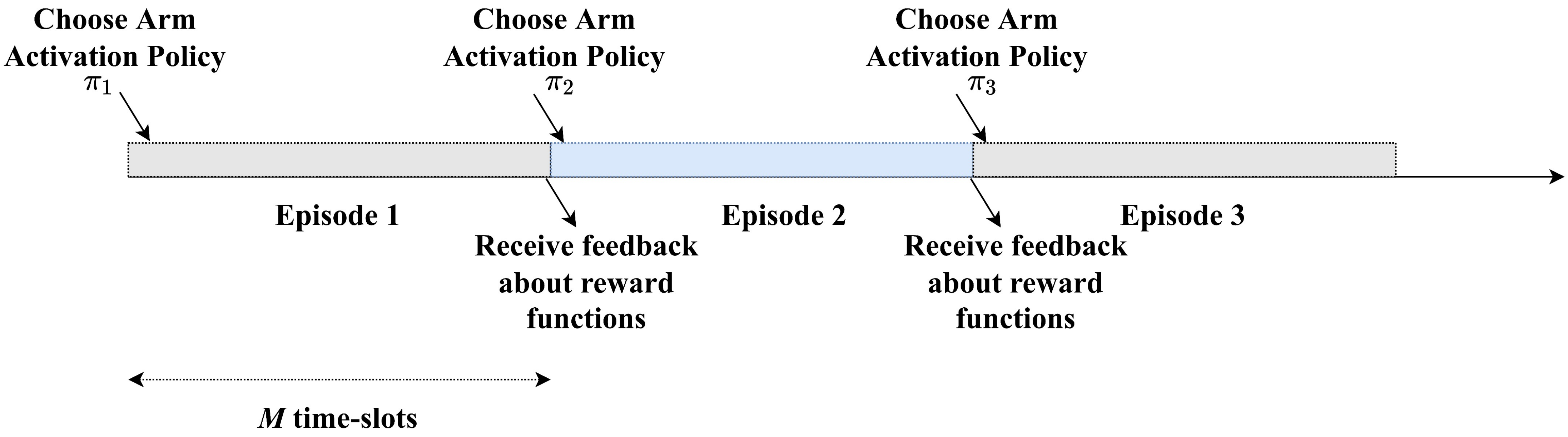
The Cost of Stale Information

- **AoI -** proxy for measuring the cost of out-of-date information

- **Major Assumption:** functions $f_i(\cdot)$ are
  1. known beforehand,
  2. remain fixed throughout,
  3. and are a good proxy for monitoring/control cost

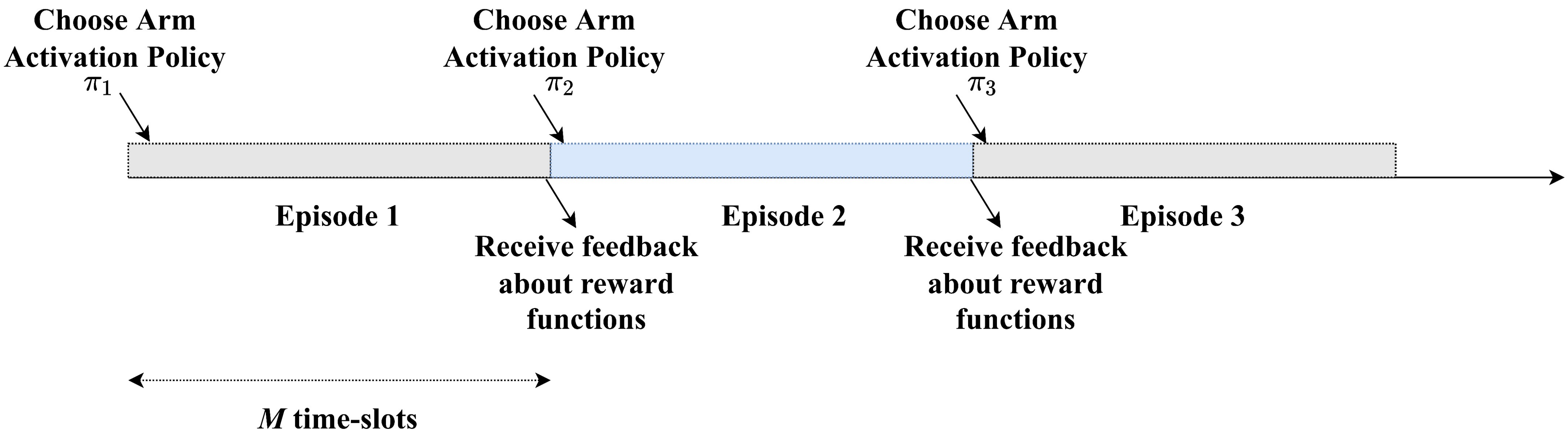
- **We ask**: What if the AoI to cost mapping is
  1. not known in advance,
  2. time-varying,
  3. and possibly adversarial?

RMAB: An Online Learning Formulation



- Episodes of length M, each episode involves solving a new AoI problem
- **AoI cost functions remain fixed** within the episode
- **Cost functions change across episodes** in an unknown manner *while maintaining indexability*

RMAB: An Online Learning Formulation



- **Q:** Can we design a scheme that learns the best scheduling policy in an online manner?

- **Answer: Yes!**

Follow the Perturbed Leader

- Recall **learning from experts** and view scheduling policies as experts:

    1. Maintain the sum of rewards observed in the past
    2. **Perturb i.i.d.** the history of rewards **for each scheduling policy**
    3. **Find the best policy** using this perturbed history

- The number of policies scales exponentially in the length of the epoch $\Theta(N^M)$

- Thus **traditional online learning methods are infeasible**

Follow the Perturbed **Whittle Index**

- **Key Idea 1:** Whittle Index acts like a low complexity optimization oracle for the RMAB problem, so incorporate it in FTPL

- **Key Idea 2:** Instead of perturbing the costs of policies, perturb the reward functions themselves

- **New Challenges** introduced:

  1. Create perturbations to **maintain indexability** structure
  2. Perturbations are **no longer i.i.d.** per expert/policy
  3. Whittle Index is an approximate but **not exact** maximizer

- **Our Contribution:** resolving these challenges!

Algorithm 2: Follow the Perturbed Whittle Leader

Input : parameter $\epsilon > 0$

Set $F_1^{(i)}(j) = j, \forall i \in \{1, ..., N\}, \forall j \in \{1, ..., M\}$

while $t \in 1, ..., T$ **do**

Set $A^{(1)}, ..., A^{(N)} = \mathbf{1}$

Sample $\delta_t^{(i)}(j) \sim$ uniform in $[0, 1/\epsilon]$, i.i.d. $\forall i \in \{1, ..., N\}$ and $\forall j \in \{1, ..., M\}$

Compute $\gamma_t^{(i)}(j) = \sum_{k=1}^{j} \delta_t^{(i)}(k), \forall i, j$

Monotone Perturbation

Choose scheduling policy

$$\pi_t = \text{Whittle}\left(F_t^{(1)} + \gamma_t^{(1)}, ..., F_t^{(N)} + \gamma_t^{(N)}\right)$$

Whittle Index Scheduling

Incur loss = $C_t(\pi_t)$ over epoch t and observe feedback on $f_t^{(1)}, ..., f_t^{(N)}$

In case of bandit feedback, construct cost estimates $\hat{f}_t^{(i)}, \forall i \in \{1, ..., N\}$ using linear interpolation

Update

$$F_{t+1}^{(i)} = \begin{cases} F_t^{(i)} + f_t^{(i)}, \forall i \in \{1, ..., N\}, & \text{if full feedback} \\ F_t^{(i)} + \hat{f}_t^{(i)}, \forall i \in \{1, ..., N\}, & \text{if bandit feedback.} \end{cases}$$

Accumulate Cost Functions

end

Regret of FPWL

Given N sources, T epochs, M time-slots per epoch and upper-bound D on cost

$$\mathbb{E}[\text{Regret}_T(\text{FPWL})] \leq \alpha T + 2D\sqrt{2MNT}$$

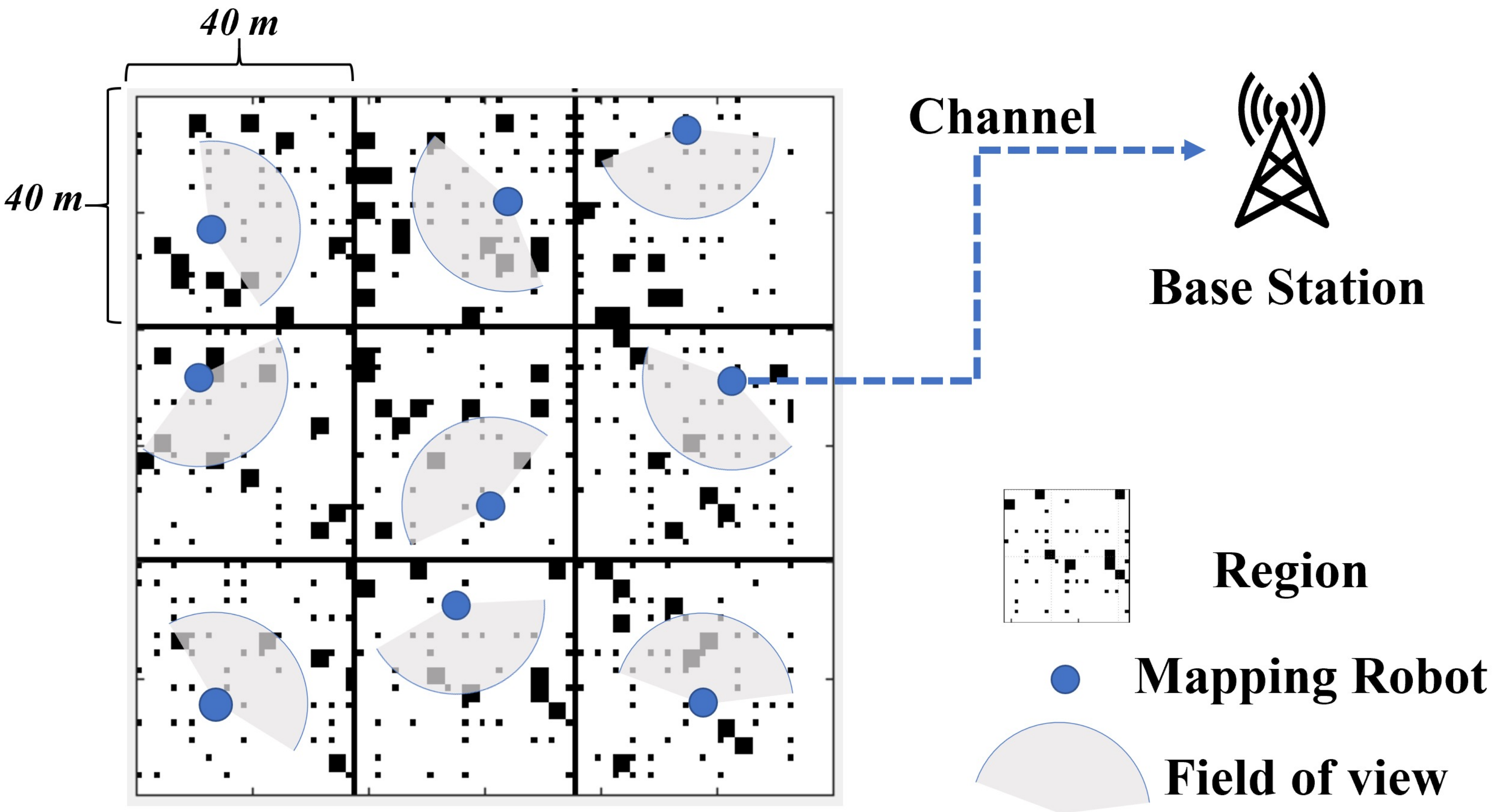
- α measures how close the Whittle-index solution is to optimality in the offline problem
- Specifically, for any two sets of cost functions f and g **assume**

$$\left| C_g\left(\text{Whittle}(f)\right) - C_g\left(\text{Opt}(f)\right) \right| \leq \alpha$$

PART IV

APPLICATIONS

Timely Occupancy Grid Mapping



- Multi-agent mapping over nine regions
- Resolution and area covered by map increases with local processing

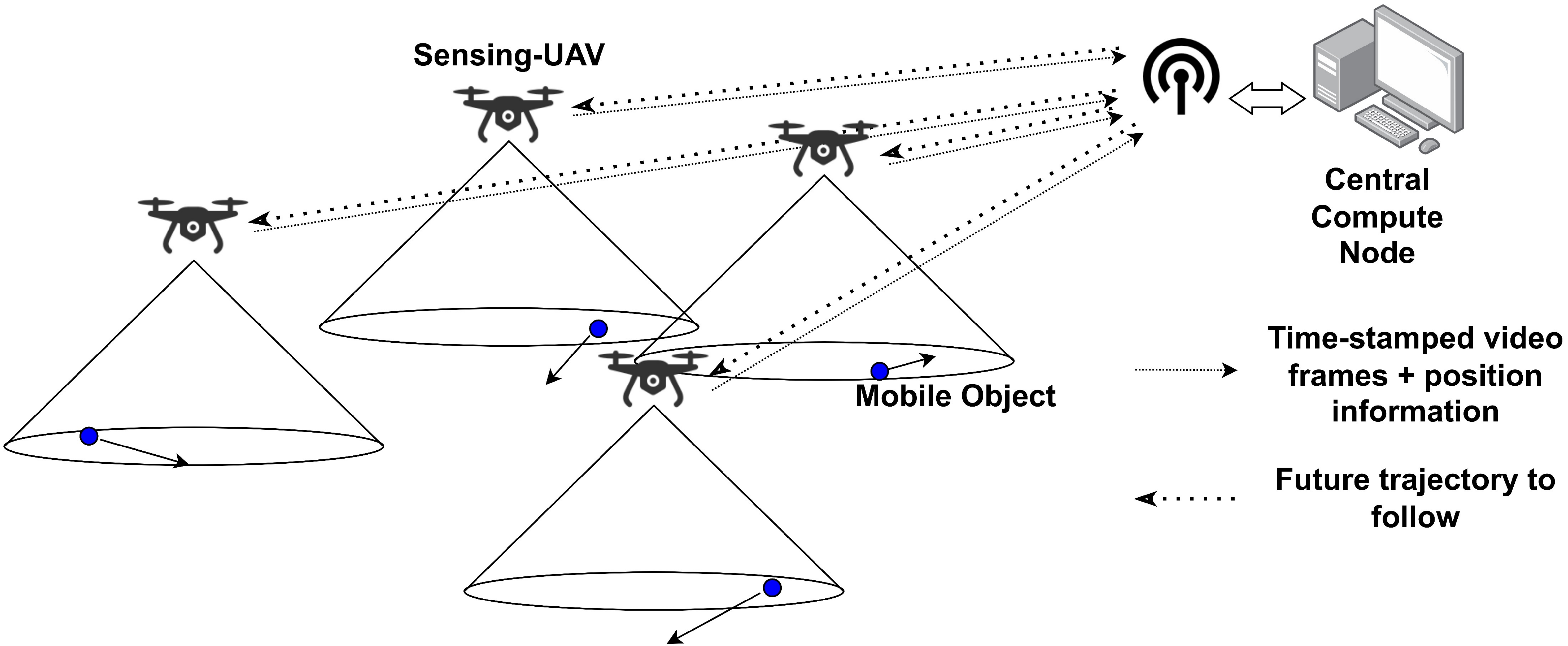
Video at

https://tinyurl.com/MultiAgentMapping

From Theory/Simulations to Implementation

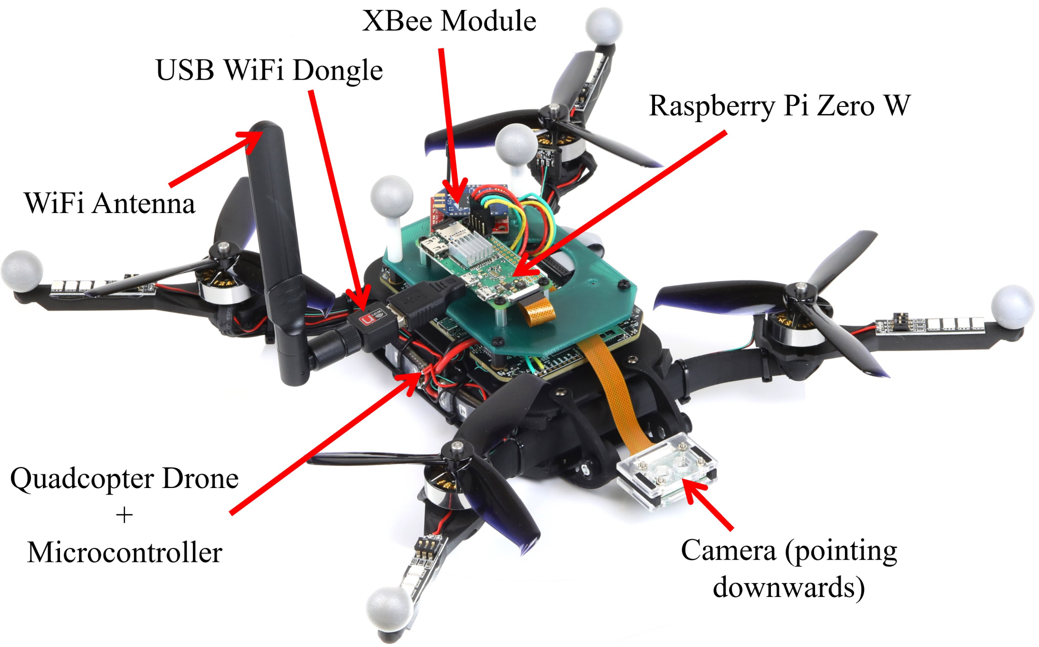
- Significant theoretical progress in AoI optimization over the last decade
- AoI was motivated by real-world monitoring and control applications
- However, system implementations have been rare
- We built a system **(WiSwarm)** to address this gap

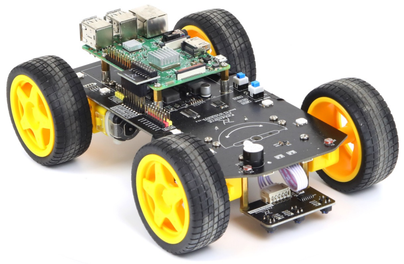
A Mobility Tracking Problem

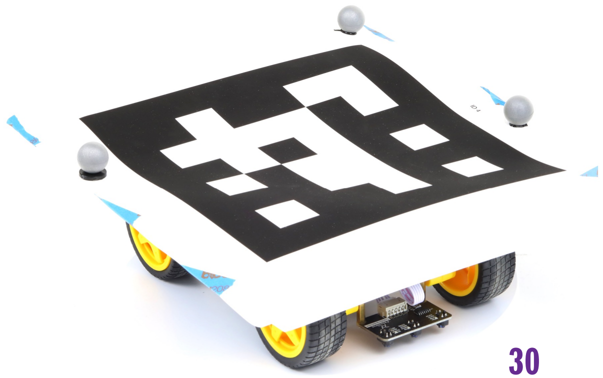


Our Setup

- Drones with cameras and WiFi but very little computation (RPi Zeros)
- Mobile cars with identifying tags that need to be tracked
- Drones collect video and send to central node for processing
- Fresh information key to good tracking

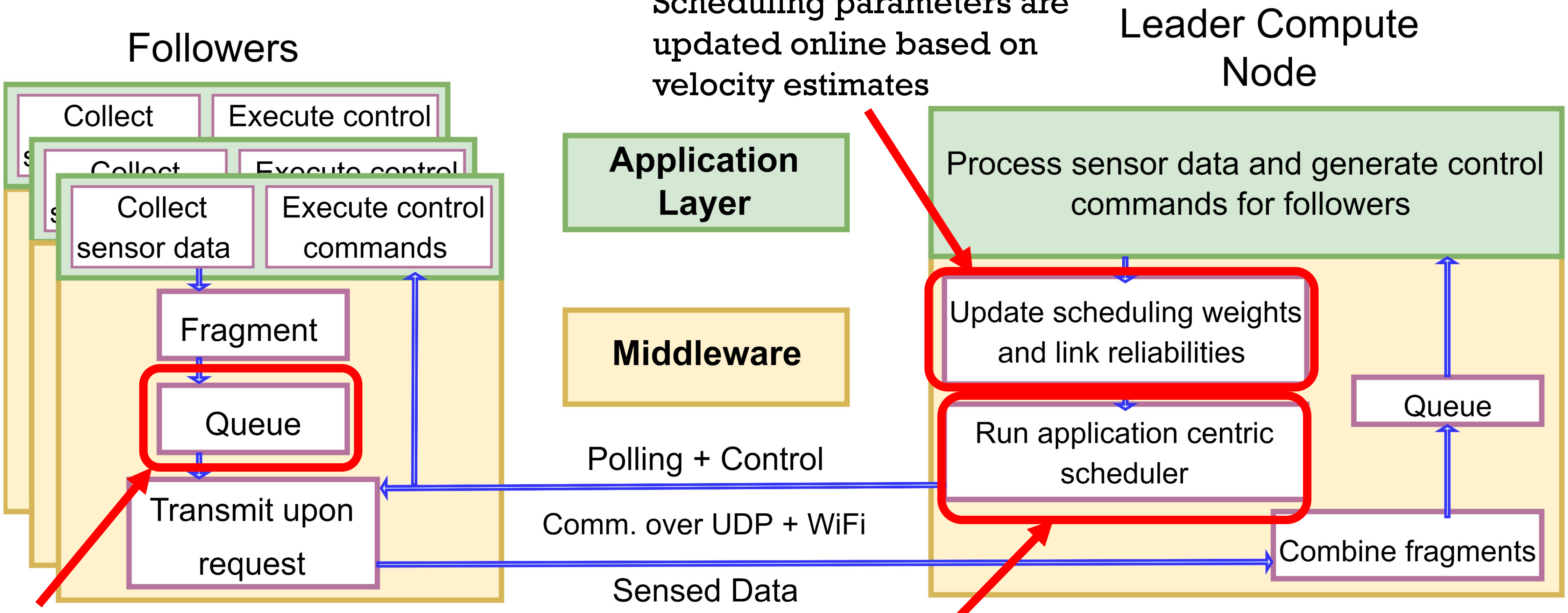






WiSwarm: An Overview

Scheduling parameters are updated online based on velocity estimates



LIFO Queues with single update buffer maintain freshness at the drones

Whittle Index scheduler decides which drone gets to transmit and prevents packet collisions

Experimental Results

- Baseline system (FIFO + WiFi UDP) - **at most 2** targets at a time
  - Avg. AoI = **0.19 seconds**
  - Avg. tracking error = **1.85 meters**
- WiSwarm (LIFO + Whittle UDP) - **at least 5** targets at a time
  - Avg. AoI = **0.16 seconds**
  - Avg. tracking error = **0.36 meters**
- Large performance improvements despite being at application layer, a MAC layer scheduler could produce even larger gains

Video at http://tinyurl.com/WiSwarm-Video

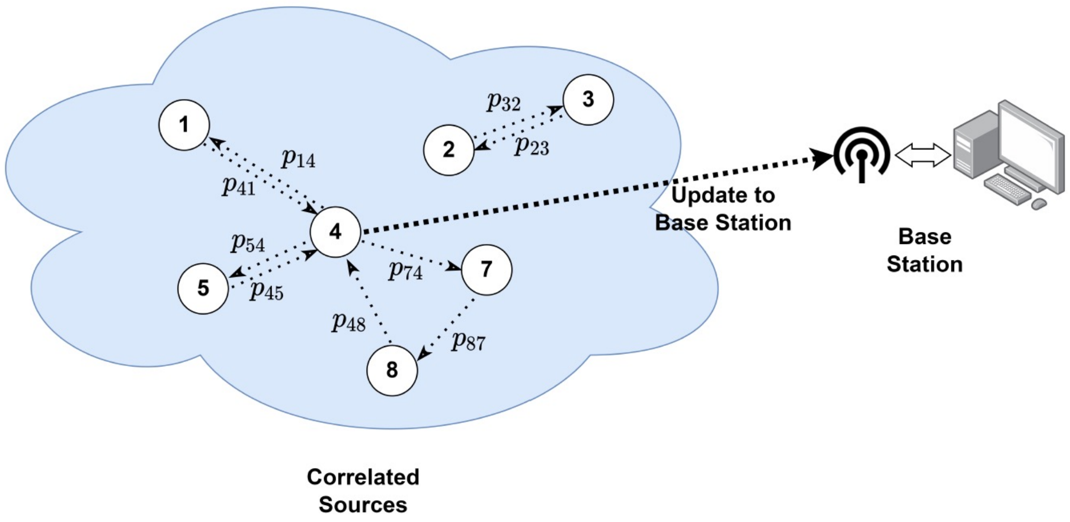
Other Works: Correlated Sources and AoI

- **Observation 1:** Prior works assume decoupled sources

- **What happens when sources are coupled or send correlated updates?**

- **Partial Answer**: For a simplified model

  1. Characterize the benefit of correlation
  2. Find policies that take correlation into account
  3. Provide performance guarantees



Other Works: Distributed Scheduling

- **Observation 2:** Prior works propose centralized policies

- **Can we provide performance guarantees for distributed policies?**

- **Partial Answer**: For weighted sum AoI

1. Standard CSMA uses i.i.d. exponential back-off timers
2. Modify back-off timer timers to be dependent on AoI
3. Provide near-optimal performance guarantees

$$Z_i(t) \sim \exp\left(\alpha^{w_i A_i^2(t)}\right)$$

Fresh-CSMA, Infocom 2023

PART V

ACKNOWLEDGEMENTS

THE END

Questions?