Application Provisioning in Fog Computing-enabled Internet-of-Things: A Network Perspective

Ruozhou Yu, Guoliang Xue, and Xiang Zhang

Arizona State University

Arizona State University

Outlines

Background and Motivation

System Modeling

Algorithm Design and Analysis

Performance Evaluation

Discussions, Future Work and Conclusions

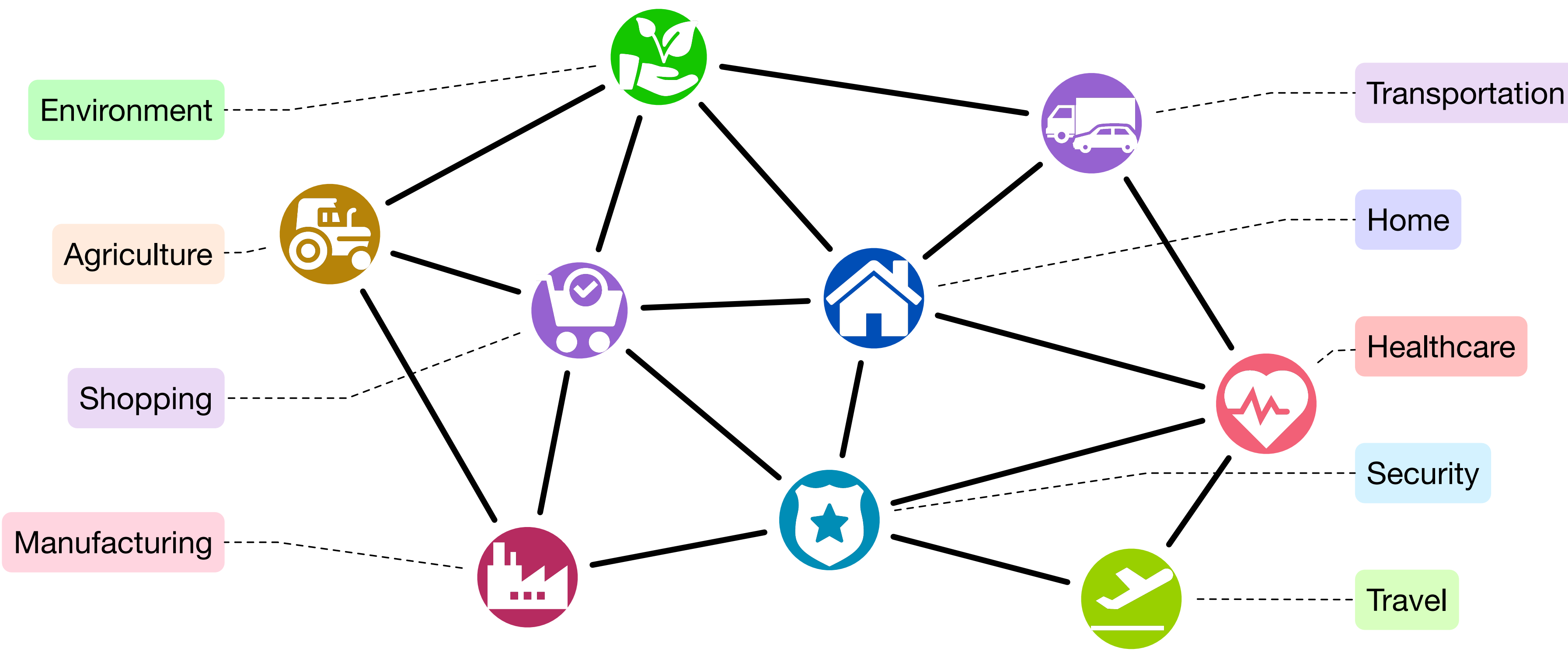
the IoT

All things are connected through ~~**the Force**~~.

— *The Jedi Faith*

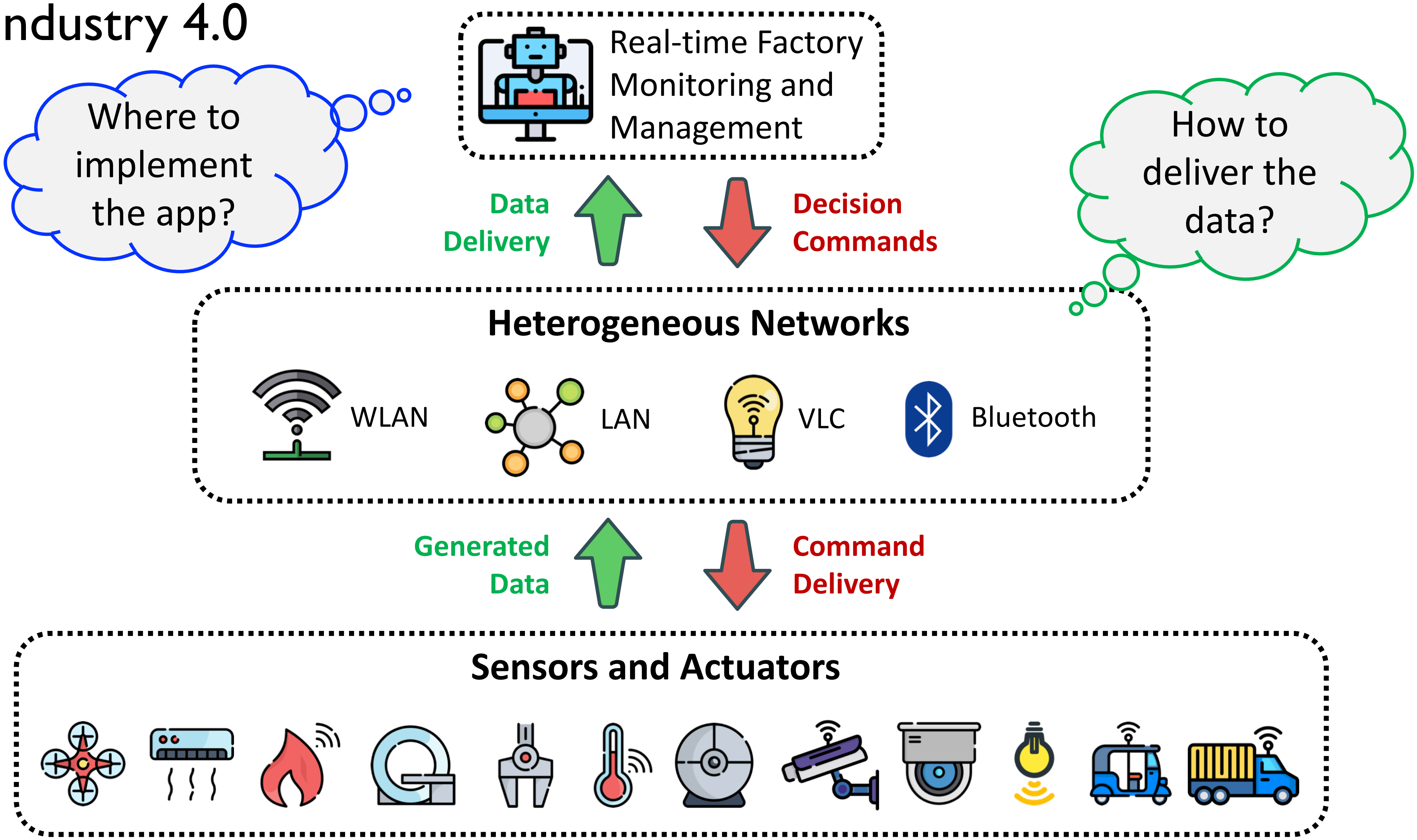
IoT: The Future Internet

- IoT is the future Internet that connects every aspect of our work and life.



A Typical Scenario in IoT

- Industry 4.0



Current Approaches

Application Hosting

Local Server:
- High CAPEX/OPEX
- Non-elastic

Cloud Computing:
- WAN congestion
- Long latency
- Unpredictable

Mobile Offloading
- one-hop
- network-agnostic

Data Delivery

ICN:
- On-demand
- QoS-agnostic
- Not real-time

TE:
- BW-oriented
- Delay-agnostic

QoS Routing:
- Single-path
- No sharing

Traditional view: **No coordination between two domains!**

Our Approach: Overview

Problem Modeling

1) Joint application hosting and data routing.
2) General graph-based IoT network model.
3) Application QoS requirements.
4) Two types of applications.
5) Inter-application resource sharing.

Algorithmic Results

1) Four variants of the problem proved NP-hard.
2) FPTASs for three variants.
3) Randomized approximation for the forth one.

Next Steps (Future Work)

1) Computation-aware provisioning.
2) Reliability and security.
3) IoT and fog economics and mechanism design.

Outlines

Background and Motivation

System Modeling

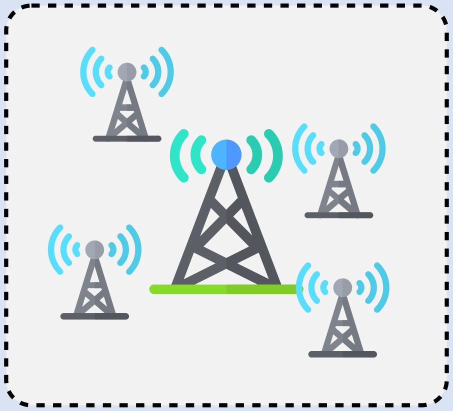
Algorithm Design and Analysis

Performance Evaluation

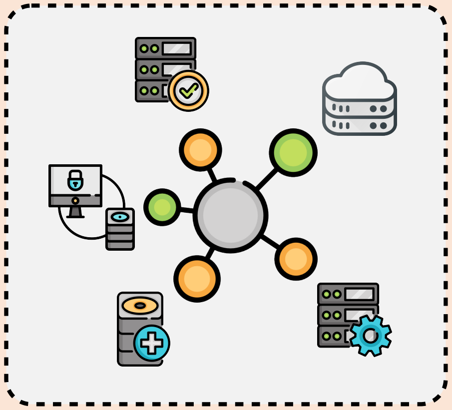
Discussions, Future Work and Conclusions

IoT Network: A General Model

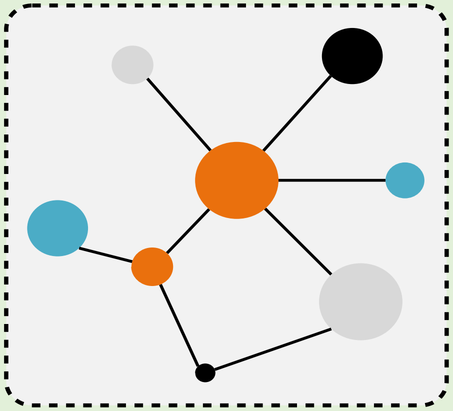
- Challenge: heterogeneous network environments



Wireless RANs:
- Geo-distributed
- Limited capacity
- Interference



Edge Network:
- Complex topo
- Distributed
- Dynamic load

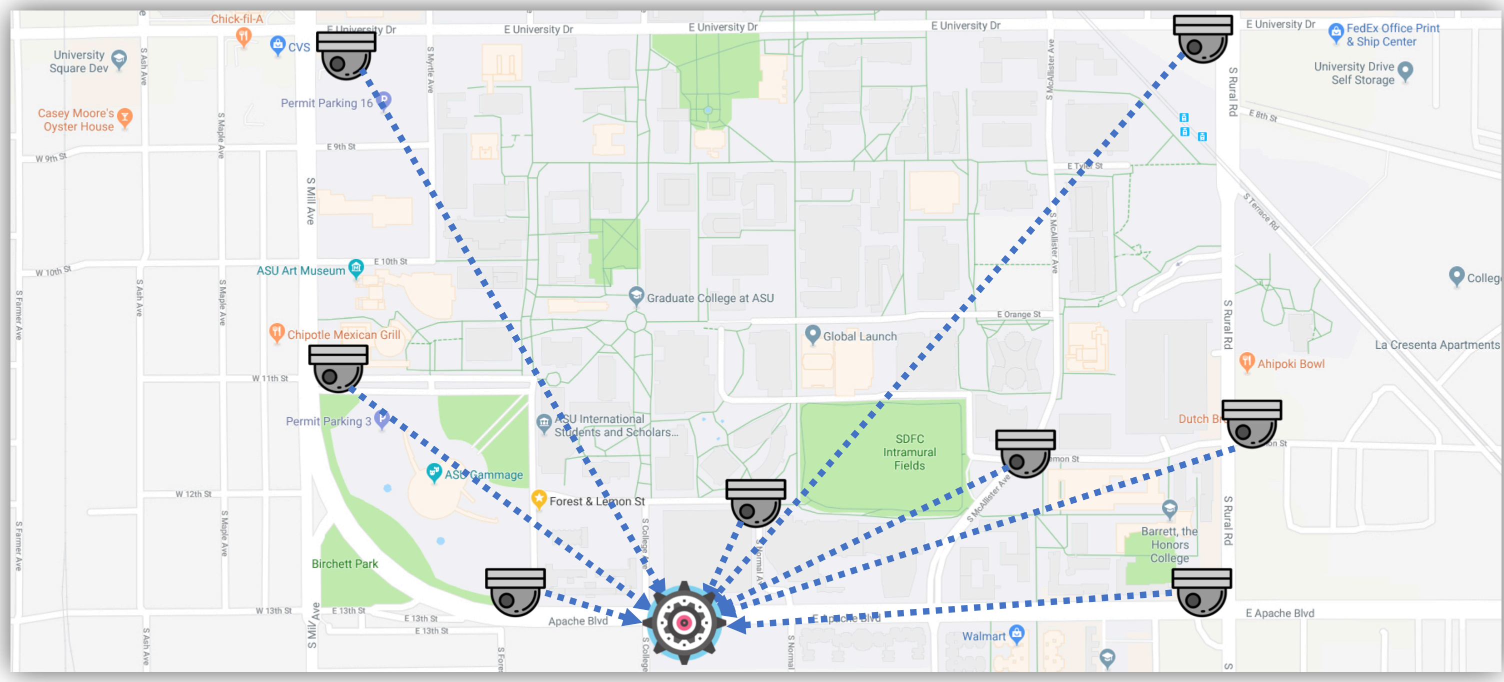


Backbones:
- Large-scale
- High latency
- ISP policies

- Model: general weighted directed graph, with some fog nodes
  - Weights: capacity & delay

Real-time IoT Applications

- Application = Logic + Data
  - Logic: data processing unit
  - Data: from multiple sources in the network

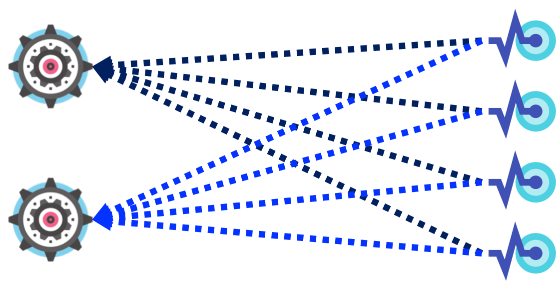


  - ***Requirements:***
    1) Bandwidth: channels supporting each data source's transmission demand
    2) Real-time: channel latency up to a required bound

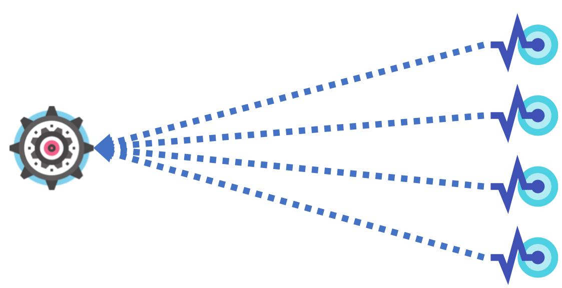
ARIZONA STATE UNIVERSITY

Two Types of Applications

- Parallelizable Applications (P)
  - Logic splittable among multiple parallel instances
  - ***Requirement:*** data in the same time interval received at the same instance
  - Example: stateless sensor data fusion



- Non-Parallelizable Applications
  - Logic has to be centrally implemented



Some icons are taken from icons8.

Two Provisioning Scenarios

- Single Application Provisioning (SAP)
  - Provisions one application at a time
  - Low complexity, suitable for general online provisioning
  - No inter-application resource sharing

- Multi-Application Provisioning (MAP)
  - Jointly provisions multiple applications simultaneously
  - Better optimization across applications, more balanced load
  - High complexity, weaker performance guarantee

Problem Statement: Overview

- **Inputs:**
  - Network topology
  - One application / Multiple applications

- **Outputs:**
  - Host designation for each application
  - Data routing for each application's each data source
    - Multi-path routing for best optimization

- **Constraints:**
  - Bandwidth demand of each application's each data source
  - Capacity limit of each link
  - Latency constraint of each application

- **Objective:**
  - Maximize **Inverse** Maximum Link Load (Load Balancing)

The Provisioning Problems are Hard!

- Four variants of the problem:

  (O- stands for the optimization version with load balancing objective)

  - **PO-SAP**: Single Application Provisioning for Parallelizable Applications
  - **O-SAP**: Single Application Provisioning for Non-Parallelizable Applications
  - **PO-MAP**: Multi-Application Provisioning for Parallelizable Applications
  - **O-MAP**: Multi-Application Provisioning for Non-Parallelizable Applications

Lemma: All four variants are NP-hard!

Proof: A simple reduction from the *MultiPath routing with Bandwidth and Delay constraints (MPBD)* problem, which is NP-hard.

Outlines

Background and Motivation

System Modeling

Algorithm Design and Analysis

Performance Evaluation

Discussions, Future Work and Conclusions

Our Results

- **Fully Polynomial-Time Approximation Scheme (FPTAS)** can achieve the best trade-off between time and accuracy
  - Approximation ratio: $(1-\epsilon)$ – For maximization problem
  - Time complexity: $O(\text{poly}(1/\epsilon) \times \text{poly}(\text{input}))$
  - In practice, one can arbitrarily tune ϵ to get best accuracy within time limit.

- Our results:

Theorem:

1) Three variants (**PO-SAP, O-SAP, PO-MAP**) admit FPTASs.
2) For **O-MAP**, there is a non-trivial approximation algorithm.

A Brief Overview of Our FPTASs

- Idea:
  - Distribute flow as even as possible
    - Push flow along under-loaded links/paths
  - Fractionalize host designation based on flows

- Approach: Primal-Dual algorithm
  - Dual lengths: exponential in primal flow values
  - Flow pushing: along dual-shortest paths
  - Flow distribution: proportional to each flow's demand
  - Stopping criteria: total dual length exceeding balancing threshold

- Analysis:
  - Flows bounded by dual lengths achieve approximately even distribution

Randomized Algorithm for O-MAP

- Randomized Algorithm:
  1) Derive fractional approximated solution for PO-MAP;
  2) Independent random host selection for each application.

- Analysis:
  - Non-trivial approximation ratio through the Chernoff bound.

Outlines

Background and Motivation

System Modeling

Algorithm Design and Analysis

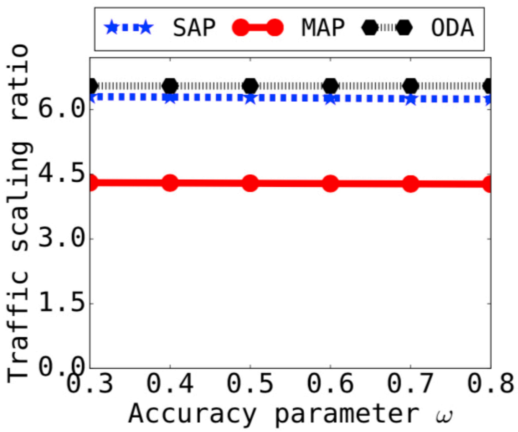
Performance Evaluation

Discussions, Future Work and Conclusions

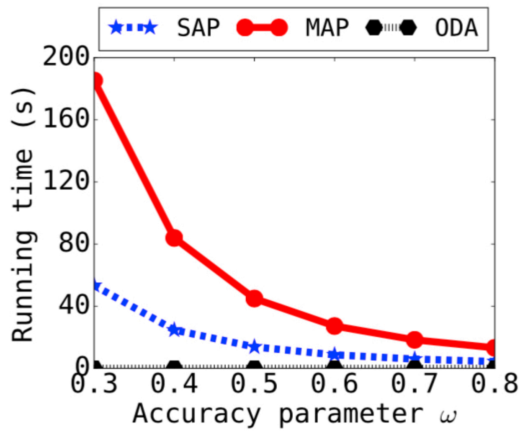
Simulation Settings

- Simulated network scenarios:
  - Random Waxman network (α=β=0.6)
    - Link capacities: [10, 100] Mbps
    - Delays: [1, 10] ms
  - 20% random fog nodes
  - 5 IoT applications
    - Data sources: [3, 10]
    - Bandwidth demands / source: [1, 25] Mbps
    - Latency bounds: [15, 25] ms
  - Approximation parameter: ϵ=0.5
- Comparisons:
  - ODA: latency-agnostic optimal solution (upper bound)
  - NS, RS: nearest / random host designation
  - GH, DA: greedy shortest-path routing / optimal delay-agnostic routing

Comparison Results



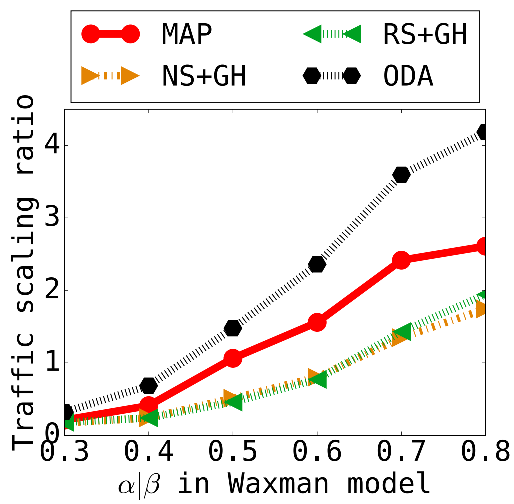
(a) Traffic scaling ratio

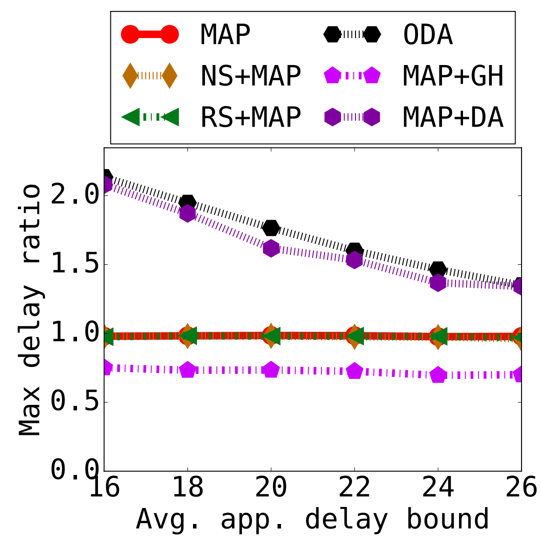


(b) Running time

With ϵ=0.5, both O-SAP and O-MAP achieves much better performance than proved bounds.

O-MAP improves upon heuristics in terms of both HD and DR, with strictly bounded delay.





Outlines

Background and Motivation

System Modeling

Algorithm Design and Analysis

Performance Evaluation

Discussions, Future Work and Conclusions

Other Perspectives and Beyond

- So far, we've talked about
  - Topology,
  - Link bandwidth and delay, and
  - Routing.

  Network Perspective

- What we haven't considered
  - Fog computing capacities,
  - Task scheduling and completion,
  - Migrations, etc.

  Computing Perspective

  - Reliability, security and privacy.

  Security Perspective

  - Incentives, pricing, and
  - Payment methods.

  Economics Perspective

- A **unified approach** is in need for fog computing-enabled IoT.

Our Conclusions

- Application Provisioning in IoT in the Network Perspective
  - General graph model for complex network environments
  - Application requirements: bandwidth and delay
  - Objective: network-wide load balancing

- Algorithmic solutions
  - FPTASs for three variants
  - Randomized approximation for the forth one

- Future directions
  - Need for unified optimization for IoT application provisioning

Thank you very much!

Q&A?

Arizona State University