Quality Aware Query Scheduling in Wireless Sensor Networks

Hejun Wu, *Qiong Luo*, Jianjun Li, Alexandros Labrinidis

WSN Applications

Quality requirements vary with applications.

The Problem

- Ignoring quality requirements of WSN applications results in low efficiency and/or low effectiveness.
  - WSNs are resource limited.
    - Computation, communication, battery
  - Satisfaction is relative to quality requirement.
  - Latency and data quality affect each other.

Quality-Aware Scheduling

- Common goal: system profit maximization
  - Quality of Service (QoS)
    - Value over Relative Deadline [Haritsa 1993]
    - Borealis QoS model [Abadi 2005]
  - Quality of Data (QoD)
    - Update scheduling for Web QoD [Labrinidis 2001]
  - Quality contracts
    - Integrate QoS and QoD [Qu 2006]

Scheduling and Quality Awareness in WSNs

- Quality-unaware scheduling
  - FPS [Holt 2004], SS [Sichitiu 2004]
  - DCS [Wu 2006], AHS [Wu 2007]
- Quality-aware query processing
  - Sample interval as QoD [Amirijoo 2007]
  - Normalized delay as QoD [Yates 2007]
  - QoD guided load shedding [Peng 2007]
  - QoD aware active node selection [Ren 2007]

Our Work

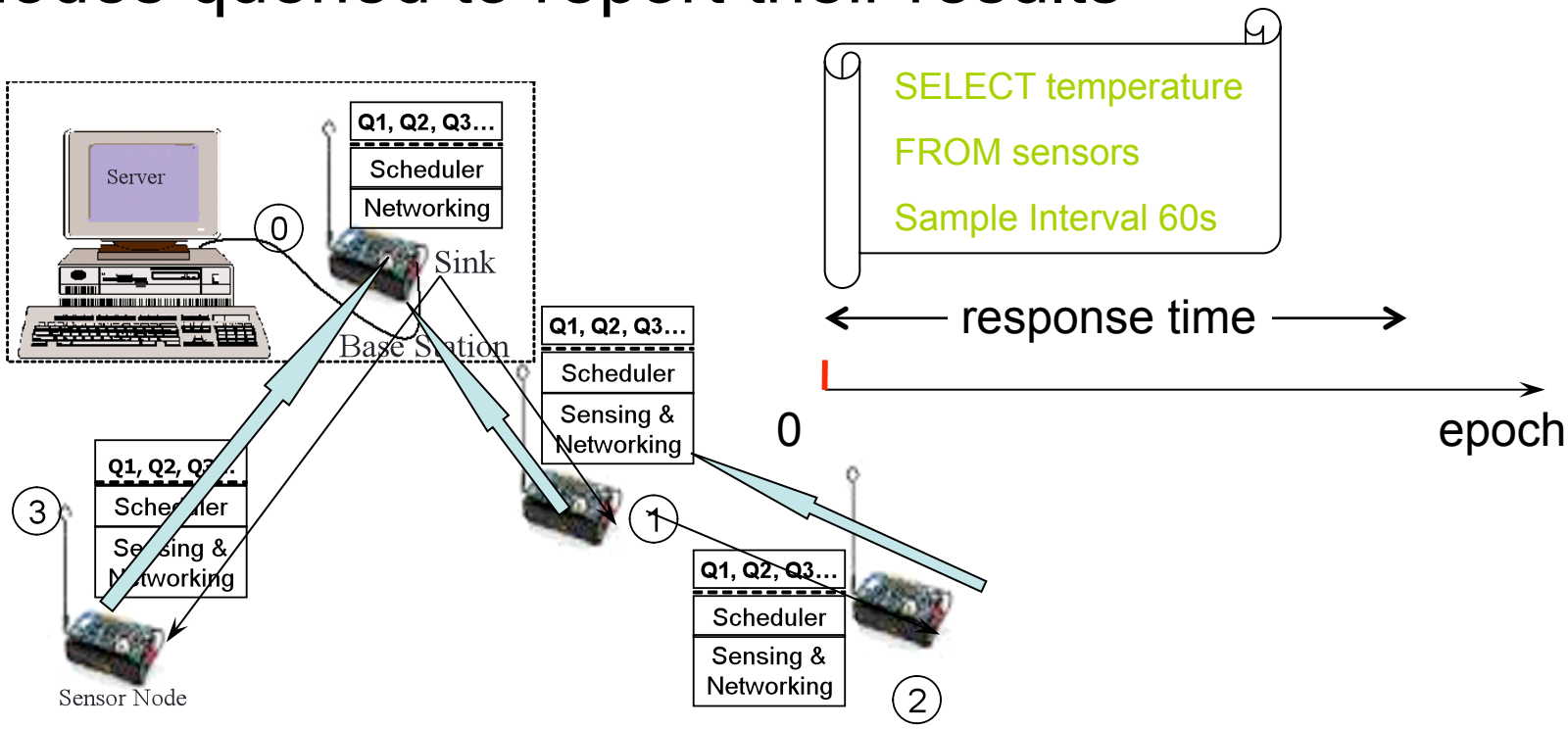
- Provide a quality-aware scheduling framework for query processing in WSNs
  - Allow users to specify quality functions
  - Work with existing quality-unaware scheduling protocols
  - Find execution order of queries in the system to maximize system profit for the underlying quality-unaware scheduling protocol

Outline

- Introduction
- Quality functions and system profit
- QAS framework
- Evaluation
- Summary

Quality of Service (QoS)

- Response time: elapsed time in one epoch for all nodes queried to report their results



Quality of Service (QoS)

- Query life time: elapsed time in number of epochs that a query runs
- QoS can be query response time, life time, or a combination of both.

Quality of Data (QoD)

- QoD $$D_Q = \frac{D}{A} - \frac{1}{H}\zeta \sum_{i=1}^{H} (\frac{D}{A} - \frac{D(i)}{A(i)})$$
  - D: #results received at the sink
  - A: All query results generated in the WSN
  - D(i): # transmitted query results at hop i
  - A(i): # results generated at hop i
  - H: Maximum # hops in the WSN

Both #total results and #results at each hop count.

Quality Award Functions

- Specified by the user, assumed linear
  - e.g., AwardtoQoD = $k3D_Q + b3$
- Total award sums up QoS and QoD.
  - Award = AwardtoQoS + AwardtoQoD

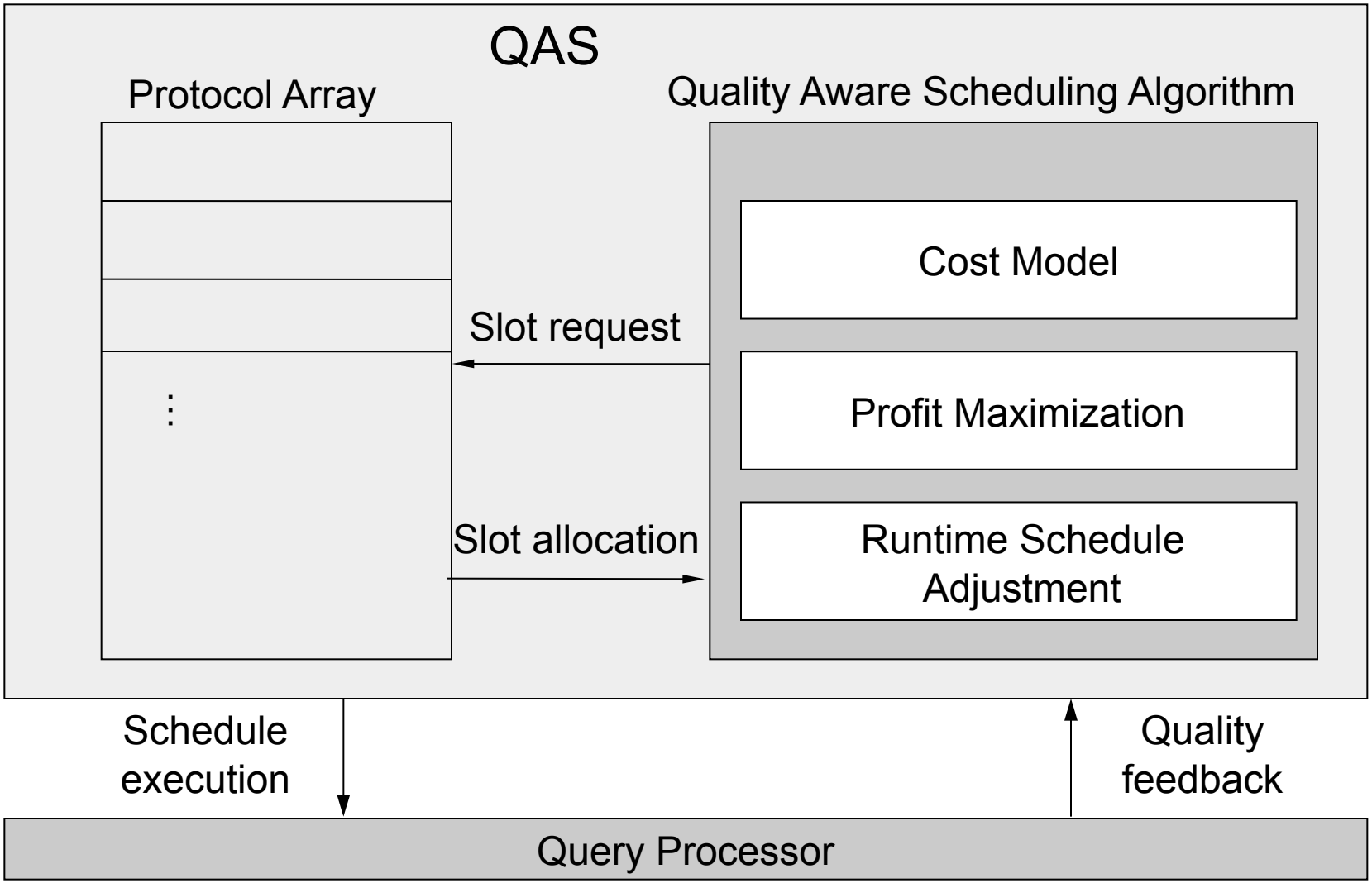
System Profit

$$Profit = Award / \varphi cost$$

φ: price of battery energy ($/joule)

Cost: total energy consumption (in joules) of scheduling and query processing

QAS Framework



QAS Overview

- Objective
  - Maximize system profit
    - Use minimum resources to maximally satisfy the application requirements
- Approach
  - Model the query processing time and energy consumption as functions of QoS and QoD
  - Determine the target quality at which the WSN can get the maximum system profit

More Details on QAS

- Cost model
  - Query execution cost + communication cost
  - *Cost* = f*(QoS, QoD)*
- Profit maximization
  - System profit P = Award(QoS, QoD) / f*(QoS, QoD)*
  - Calculate the partial derivatives of *P t*o find the QoS and QoD that maximize P
  - Determine the order or multiple queries that maximize the total P of all queries

Cost Model

- Response time

$$T_{Np} = \alpha D_Q + \beta$$

- Energy

$$E = (\lambda D_Q + \delta) L$$

Single-Query QAS

- Find the target quality that maximizes the system profit:

$$P = (k1*L+b1 + k2*T_{Np} + b2 + k3*D_Q + b3)/E$$

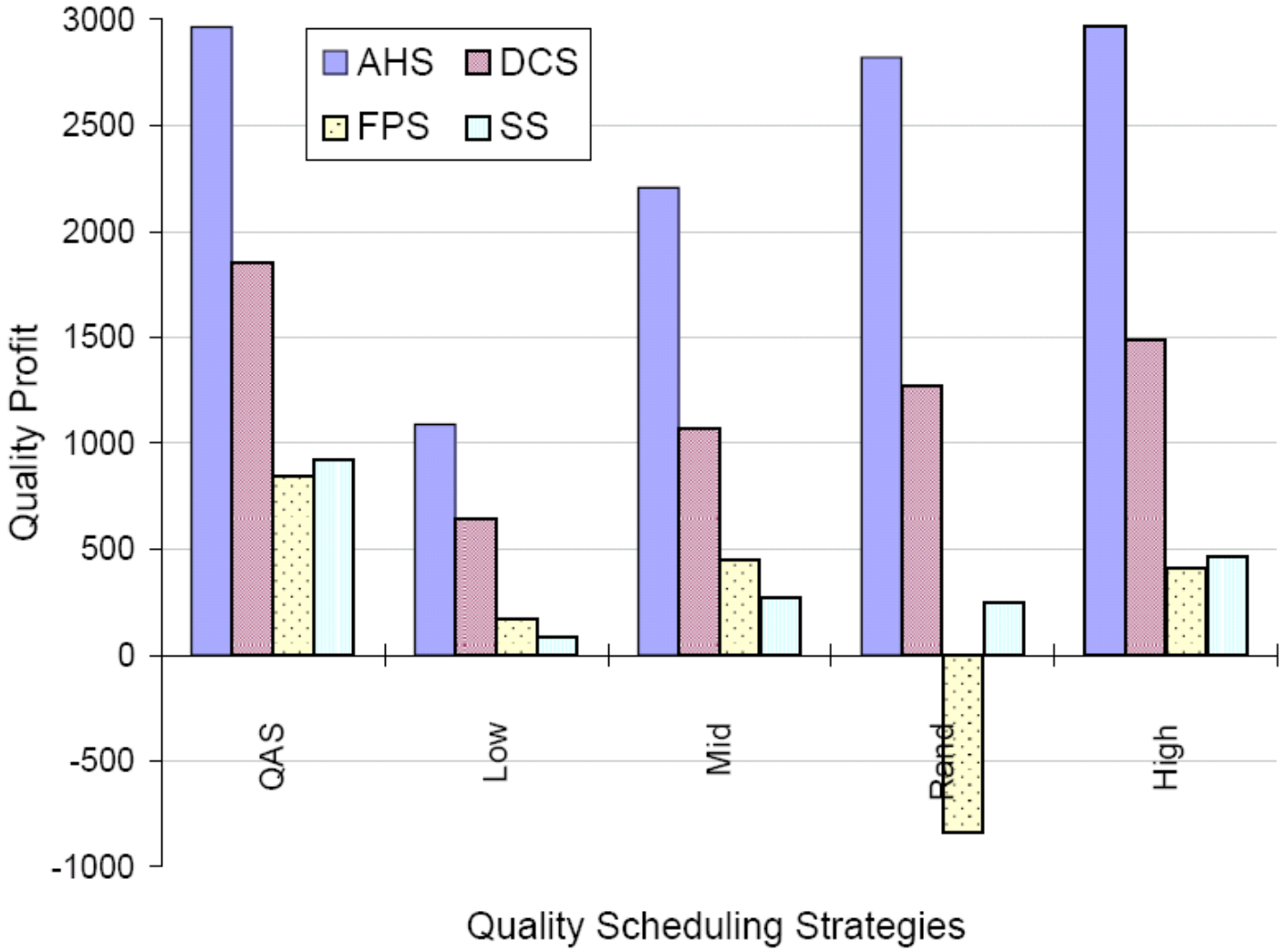
Multi-Query QAS

- Problem
  - Query execution order affects attained qualities of individual queries.
- QAS heuristic approach
  - For each query i, get its maximum profit maxP[i] as if it is the only query in the system
  - Sort queries in the descending order of maxP[i] and calculate the actual profit P
  - Sort queries in the descending order of maxP[i]/ targetQoS[i] and calculate the actual profit P'
  - Choose the order that results in a larger profit

Evaluation Setup

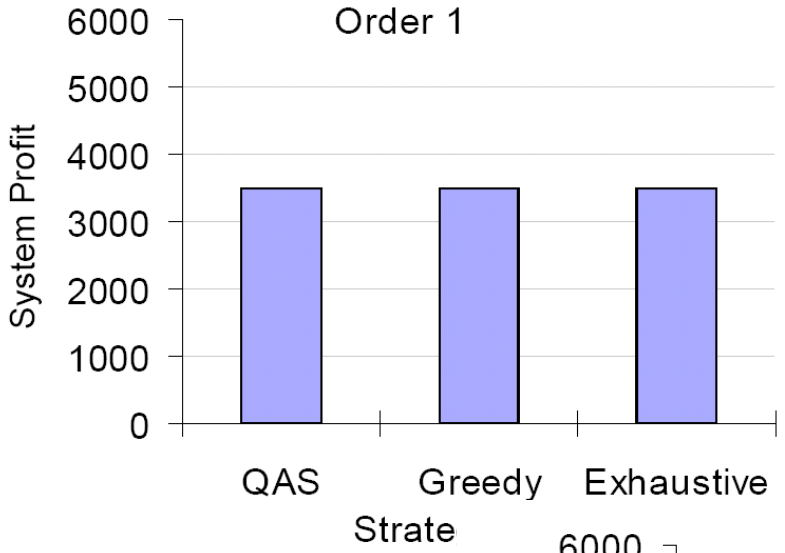
- Simulated networks
  - 100 nodes random deployed in a 100m*100m area
  - Intel-lab dataset
- Underlying scheduling protocols
  - FPS, SS, DCS, and AHS
- Quality scheduling strategies
  - Low-quality (low)
  - Medium-quality (Mid)
  - Random-quality (Rand)
  - High-quality (High)
  - Quality-Aware Scheduling (QAS)

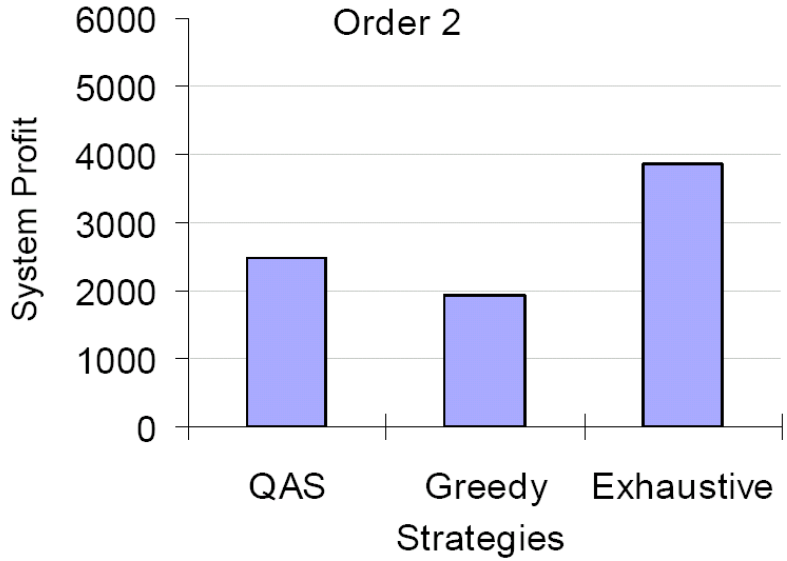
Quality Profits from Different Underlying Protocols

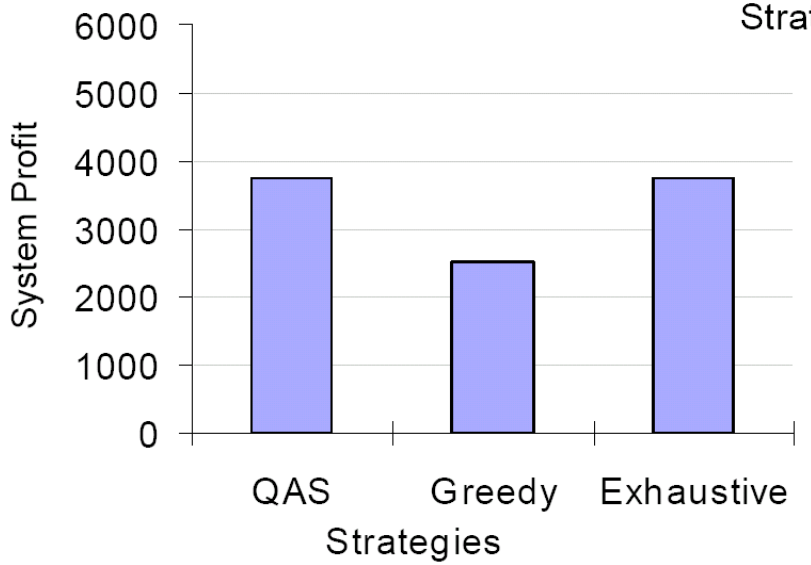


Different Quality Functions

Multiple Queries







Summary

- Quality-aware scheduling is necessary in sensor query processing
  - Applications have different quality requirements.
  - Ignorance of quality results in low efficiency.
- QAS provides a framework to enable different quality requirements of queries
  - Quality-cost estimation for individual queries
  - Greedy query execution ordering for multiple queries