



Towards Event Source Unobservability with Minimum Network Traffic in Sensor Networks

Yi Yang

Joint work with Min Shao,

Sencun Zhu, Bhuvan Uргаonkar, and Guohong Cao

Department of Computer Science and Engineering

The Pennsylvania State University

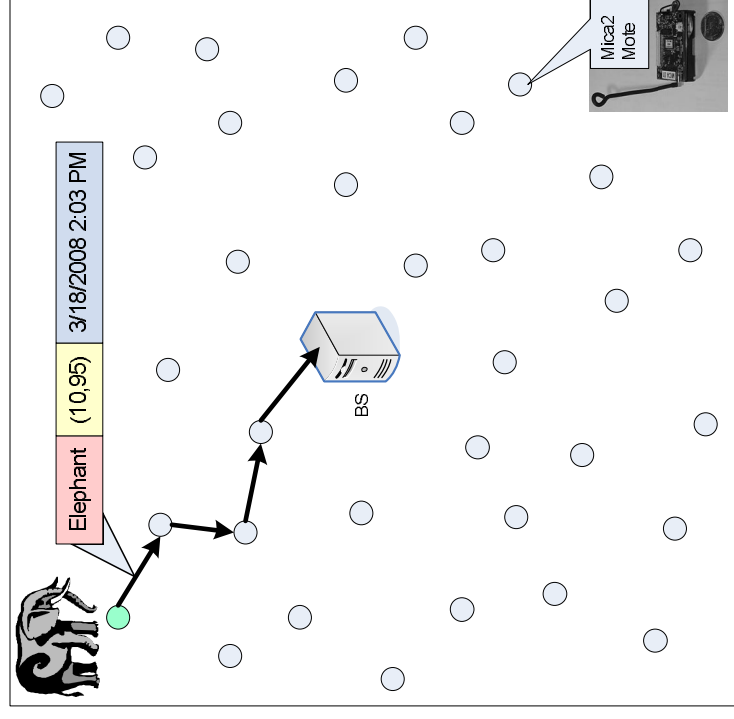
ACM WiSec 2008



Sensor Network Privacy

- Event detection & report
 - Whether, when and where an event happened is leaked to the attacker (passive monitor)
 - Even if event report packets are encrypted

- Attacker may go and capture the endangered animal!





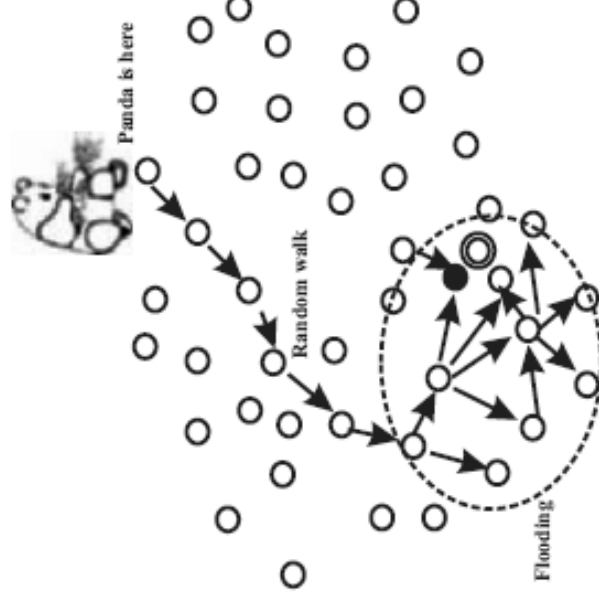
Challenges in Sensor Network Privacy

- Sensors have limited resources in energy, communication, computation, and storage
 - Lightweight, energy-efficient mechanisms are affordable
- Open architecture of standard wireless communication enables an attacker to monitor all network traffic
 - Its own sensor network covering entire deployment area
 - A site surveillance device with hearing range \geq network radius



Previous Work

- Anonymity techniques for general networks are too expensive to be employed in sensor networks
 - Mix, Mix Cascade, Onion Routing
- Phantom routing [C. Ozturk et al, SASN'04; P. Kamat et al, ICDCS'05]
 - Panda-hunter game
 - Random walk to a phantom source, flood/single-path routing to BS
- *Source location privacy under local attacker*
 - Limited coverage, single source, hop-by-hop traceback
 - Capture likelihood is 97%, if attacker's hearing range ≥ 3 times of sensors



- Others: technique to hide BS, path confusion algorithm



Introduction

- *Event Source Unobservability*
 - A stronger notion than source location privacy
 - Attacker can neither discern event occurrence nor find event source
- *An external, passive, and global attacker*
 - Examine packet content for source id
 - Encryption + all packets of same length
 - Trace back to source via identifying immediate source of transmission, if encrypted messages remain same
 - Re-encryption
 - Advanced traffic analysis: rate monitoring, time correlation
 - Network-wide dummy messages + event message forwarding latency



Baseline Scheme

- Network-wide dummy messages to hide event messages
 - Every sensor sends messages (real or bogus) at intervals following a certain distribution (constant- or probabilistic- rate)
 - Detection sources delay event message transmission to next inter-message interval following the same distribution
- Observations:
 - A large number of bogus messages lead to high message overhead, high channel collision, low delivery ratio of real event messages
 - Tradeoff between overhead and latency



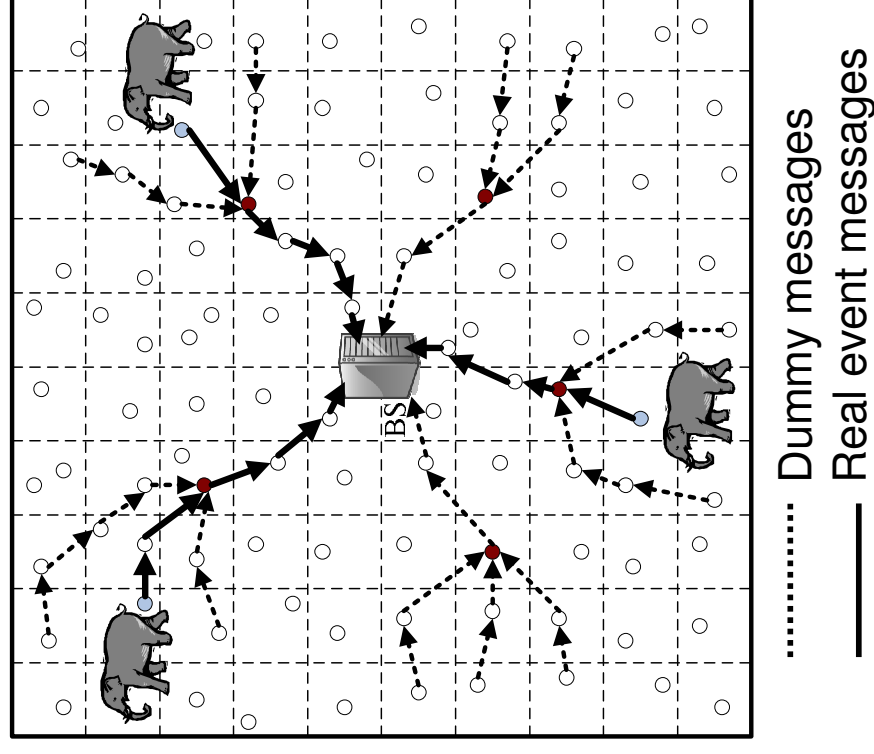
Our Contribution

- **Paramount goal: *minimize network traffic* while preserving event source unobservability**
 - Tolerate some latency
- **Two proposed schemes over the baseline**
 - Proxy-based Filtering Scheme (PFS)
 - Reduce message overhead, through proxy nodes collecting and filtering enroute dummy traffic
 - Tree-based Filtering Scheme (TFS)
 - Further reduce message overhead, because proxies closer to BS filter dummy traffic from proxies farther away



Proxy-based Filtering Scheme (1)

- How many proxies needed?
- How to place proxies to minimize network traffic?
- What is filtering operation inside each proxy?



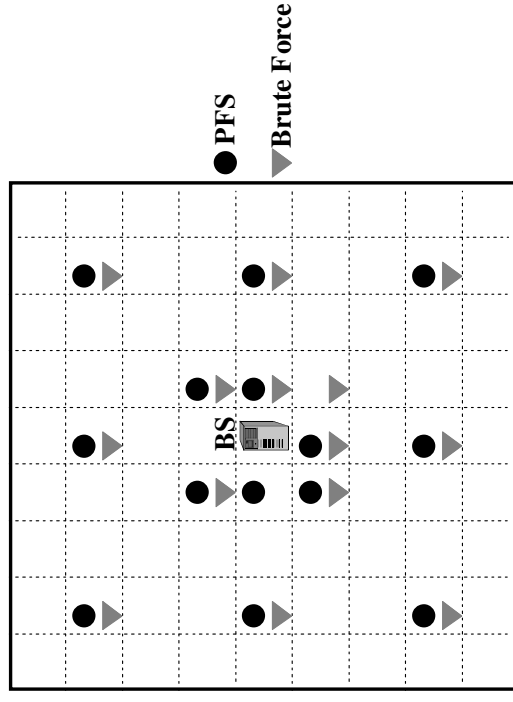
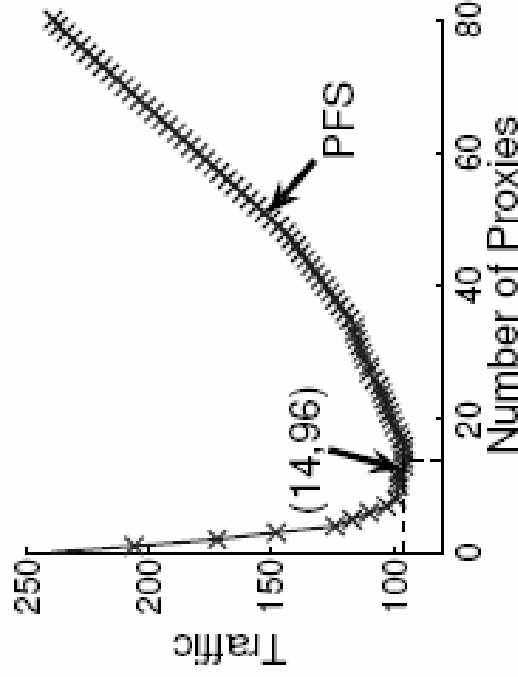


Proxy-based Filtering Scheme (2)

- Optimal proxy placement algorithm

- To minimize overhead = $r_{source} \times \sum_{i \in V} \min_{j \in P} d(i, j) + r_{proxy} \times \sum_{j \in P} c(j)$

- Modeling of the optimization: *facility location problem* (NP-hard)
- *Local search heuristics* to find an optimal solution (proxy number and placement), in an efficient way





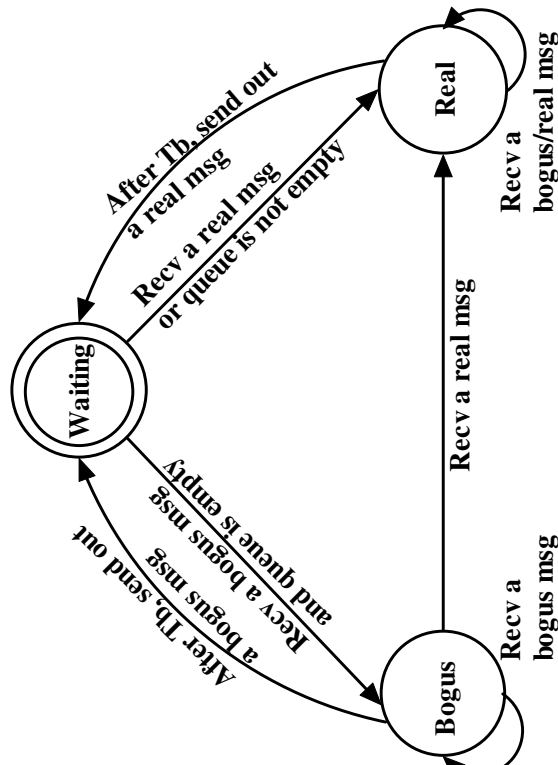
Proxy-based Filtering Scheme (3)

- After proxies are determined
 - Proxy broadcasts a HELLO message
 - Each cell responds to closest one as default proxy
 - Each cell sends encrypted bogus or real messages with exponential distribution interval towards proxy via GPSR



Proxy-based Filtering Scheme (4)

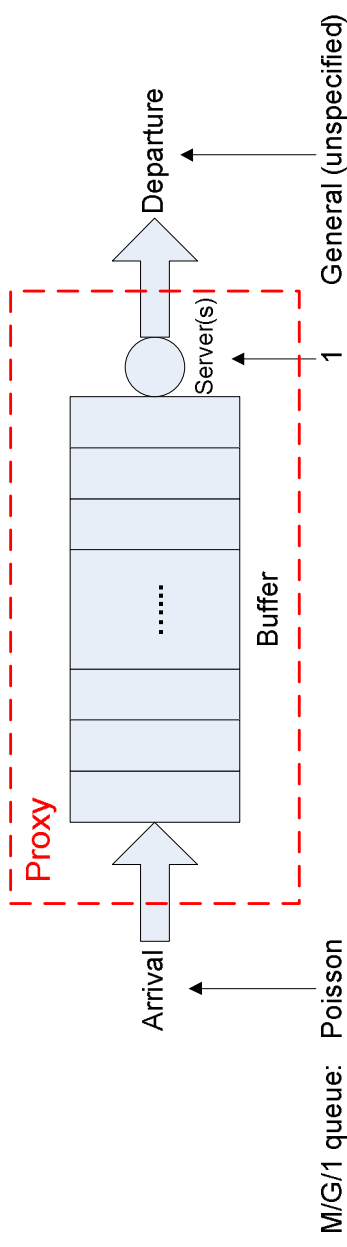
- Proxy filtering
 - Decrypt messages
 - Drop bogus messages
 - Put re-encrypted real event messages into buffer
 - After a constant time T_{proxy} , either a bogus or real message (FIFO) is sent out, depending on whether buffer is empty
- Event source unobservability is preserved





Proxy-based Filtering Scheme (5)

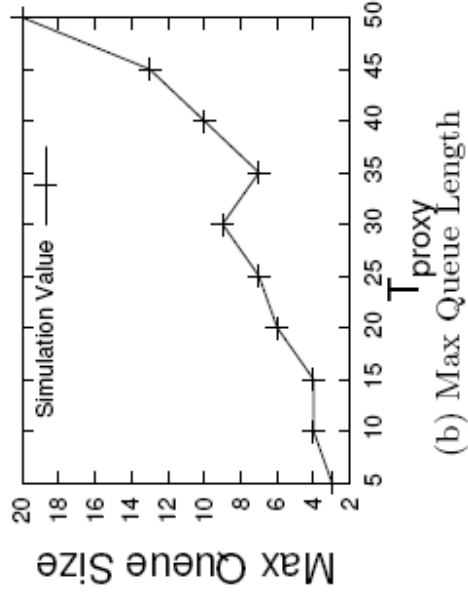
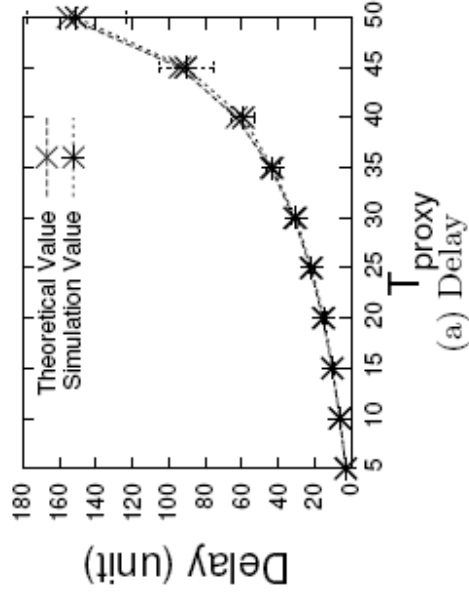
- Queuing analysis on buffering delay
 - M/G/1 queue



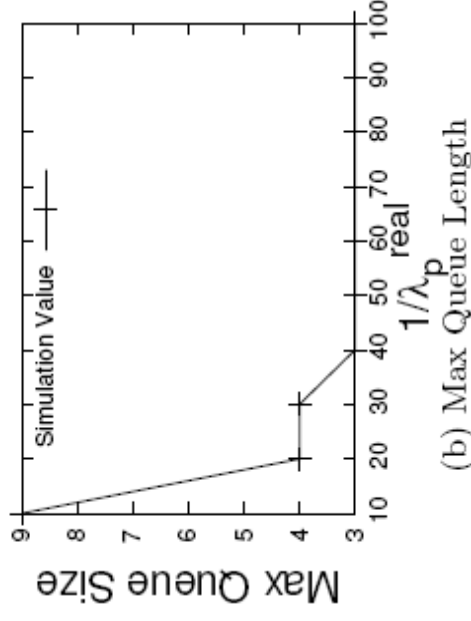
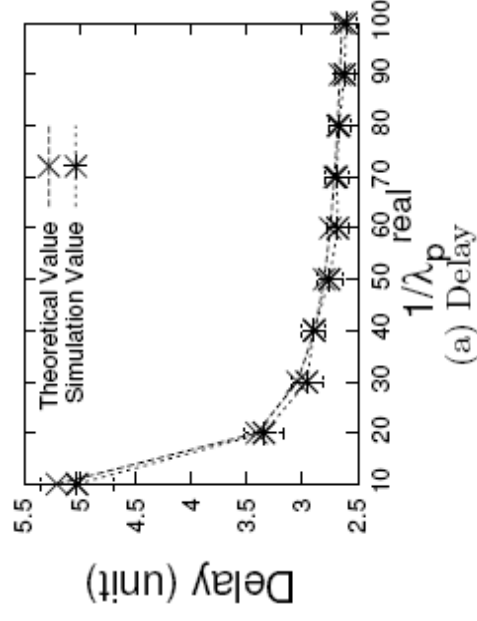
- PASTA (Poisson Arrivals See Time Averages) property
 - For Poisson Arrival, the fraction of requests finding the queue idle upon arrival equals to the fraction of time the system is idle
 - Service time s_p : $E[s_p] = p_p \frac{T_{proxy}}{2} + (1 - p_p) T_{proxy}$
- Average sojourn (waiting+service) time
 - $d_p = E[s_p] + \frac{\lambda_p^{real} (E^2[s_p] + E[s_p^2])}{2(1 - \lambda_p^{real} E[s_p])}$, increase with T_{proxy} and λ_p^{real}



Proxy-based Filtering Scheme (6)



Delay and max queue length under $\lambda = 1/60$ per time unit

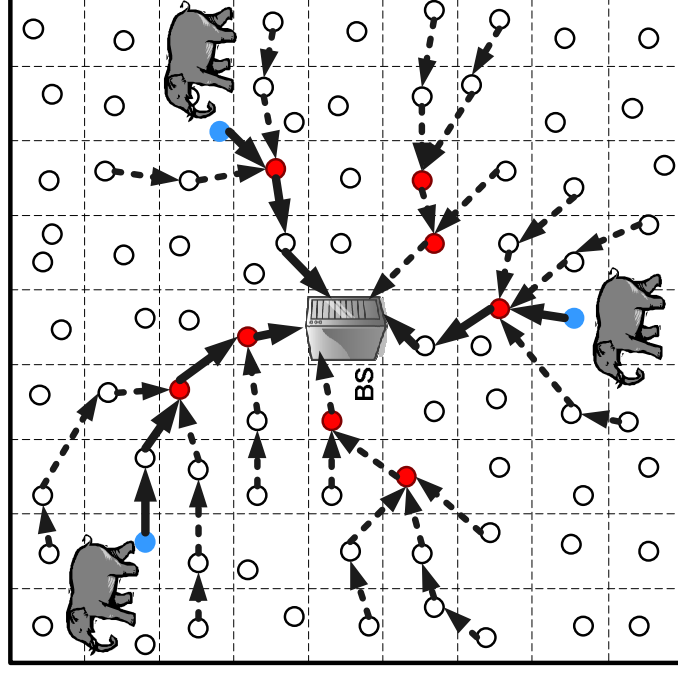


Delay and max queue length under $T_{\text{proxy}} = 5$ time units



Tree-based Filtering Scheme (1)

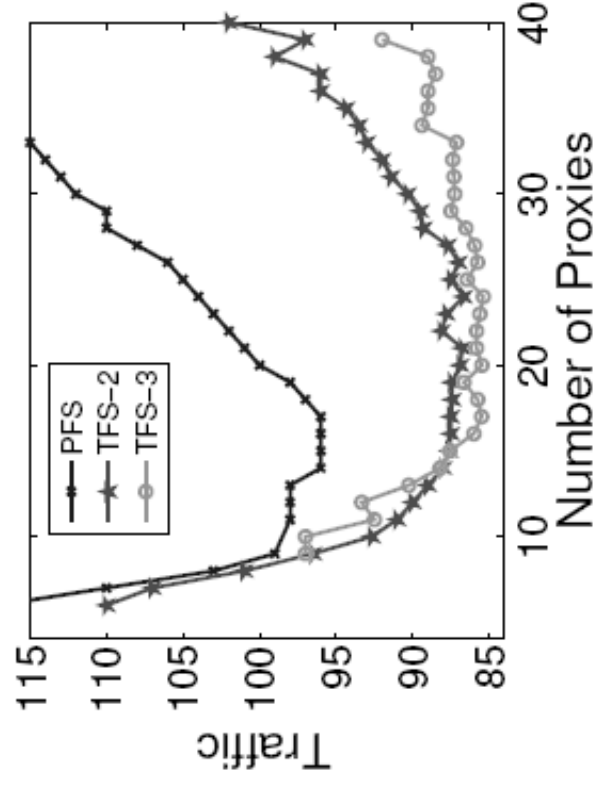
- Proxies closer to BS filter dummy traffic from proxies farther away
- Proxies form a *hierarchy* rooted at BS





Tree-based Filtering Scheme (2)

- Lower network traffic
 - More dummy messages are dropped before reaching BS
- Higher event report latency
 - Real event messages traverse multiple proxies, each of which has a buffering delay





Practical Considerations

- System parameter selection
 - $r_{\text{dummy}} \approx r_{\text{source}}$, to minimize overhead and latency at sources
 - $\overline{T_{\text{proxy}}} >$ aggregate incoming real event rate, so that real event messages are not dropped at proxies
- Role shifting of proxy nodes
 - Shift within cells coordinated by cell head
 - Shift among cells coordinated by BS (hand-off problem)



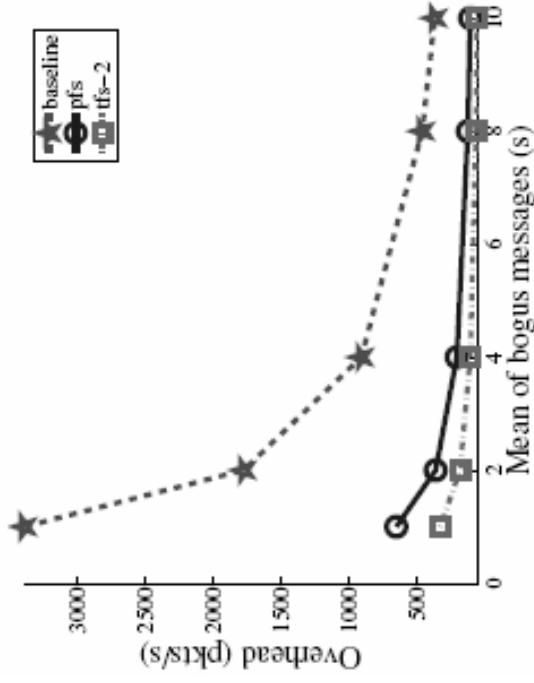
Performance Evaluation (1)

- Simulation Setup
 - Glomosim
 - 625 sensors distributed in 1000m*1000m area
 - Transmission range: 50m
 - 5 cells randomly selected as real sources
 - Heavy-rate real event msg mean: 10s, light-rate real event msg mean: 400s
 - T_{proxy} is 5s, buffer size is 10

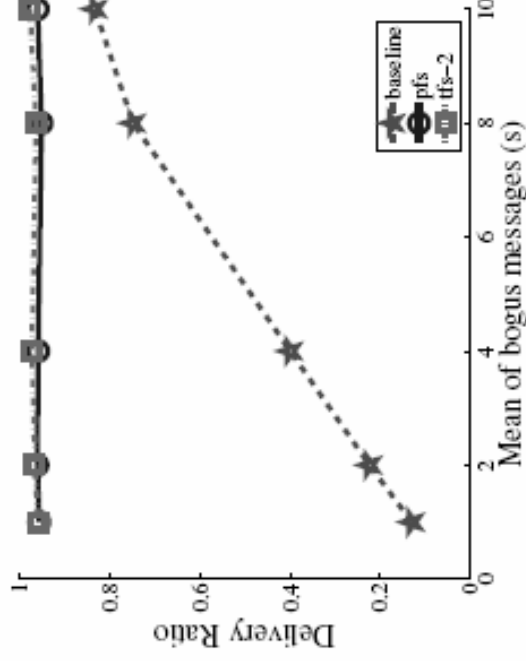


Performance Evaluation (2)

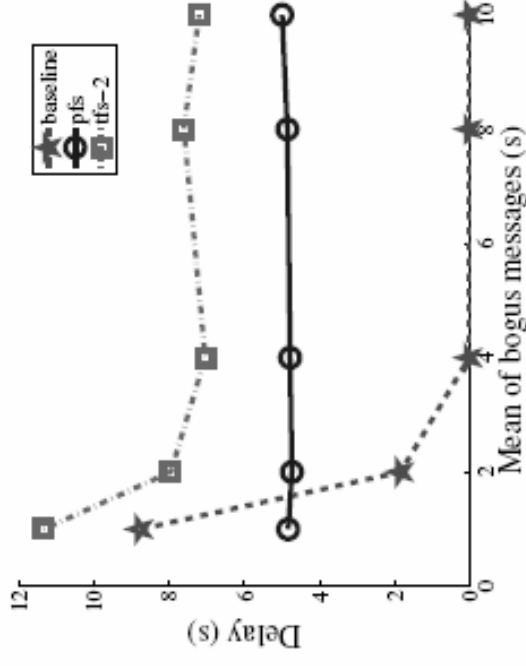
- Simulation results under heavy-rate real events



(a) Overhead



(b) Delivery Ratio

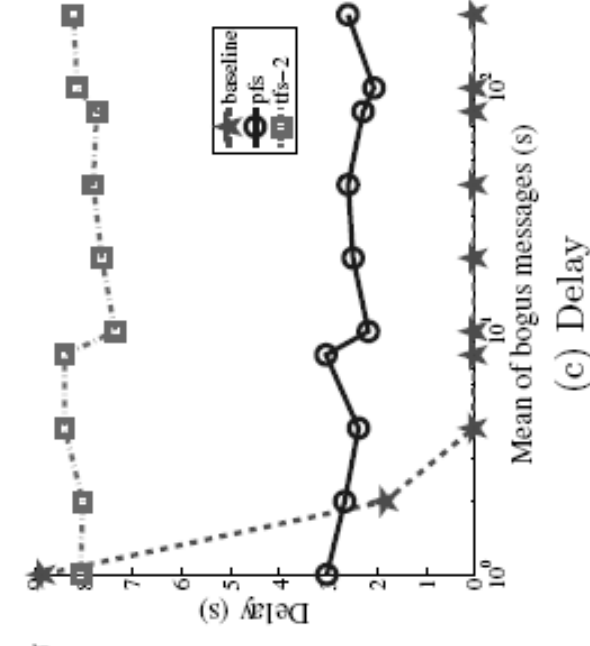
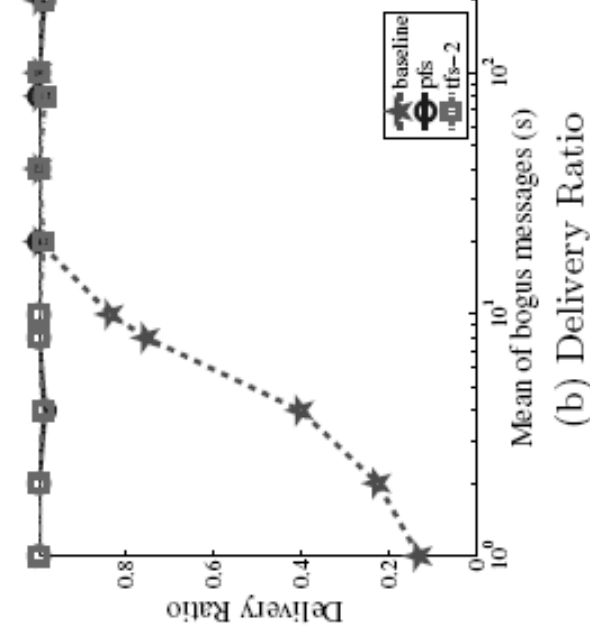
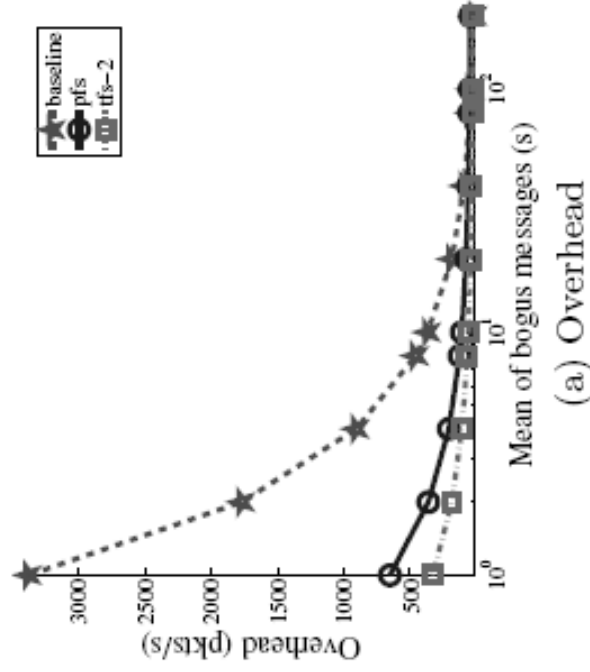


(c) Delay



Performance Evaluation (3)

- Simulation results under light-rate real events





Conclusion and Future Work

- PFS and TFS to achieve event source unobservability with minimum network traffic for sensor networks
- Optimal proxy placement algorithm via local search heuristics
- Queuing analysis on buffering delays
- Future work: extend TFS scheme
 - Optimize buffer time & size for each proxy level



Thank you ...

Questions?