

# Application-Specific Modelling of Information Routing in Wireless Sensor Networks

Bhaskar Krishnamachari and John Heidemann

Presented by David Yates

February 13, 2004

# Acknowledgments

- My virtual red team
  - Jim Kurose and Peter Desnoyers
- Companion paper
  - *John Heidemann, Fabio Silva, and Deborah Estrin, “Matching Data Dissemination Algorithms to Application Requirements”. In ACM SenSys, November 2003.*  
<http://lecs.cs.ucla.edu/publications/papers/p128-heidemann.pdf>
- Errata
  - Equation (2) in the paper has typos

# Outline

- Directed diffusion -> Taxonomy of algorithms
- Workload and network model
- Example of analysis
- Results
- Summary from paper
- Critique
  - What I liked (or was strong) in the paper
  - What I didn't like (or was weak) in the paper
  - Possible future directions

# Brief Taxonomy of Diffusion Algorithms

## **2-phase pull (aka Directed Diffusion)**

- Sink sends interest\* (every interest interval)
- Source sends exploratory data\* (every exploratory interval)
- Sink sends positive reinforcement (response to exp. data)
- Source sends data (rate defined by app.)

## **1-phase pull**

- Sink sends interest (every interest interval)
- Source sends data

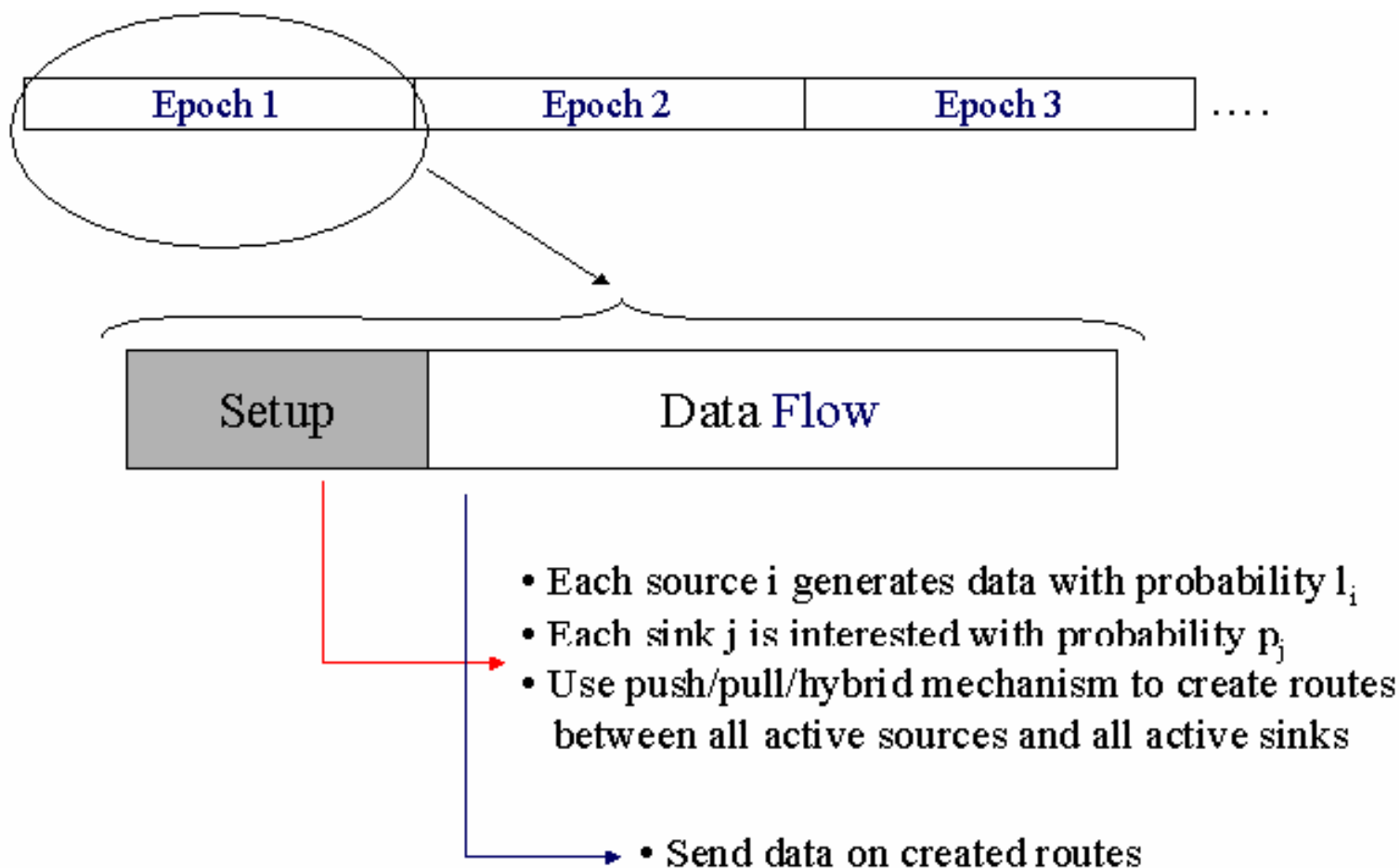
## **Push**

- Source sends exploratory data\* (every exploratory interval)
- Sink sends positive reinforcement (response to exp. data)
- Source sends data

Note (\*) indicate messages that are sent to all nodes (flooded or geographically scoped). All algorithms also have negative reinforcement messages.

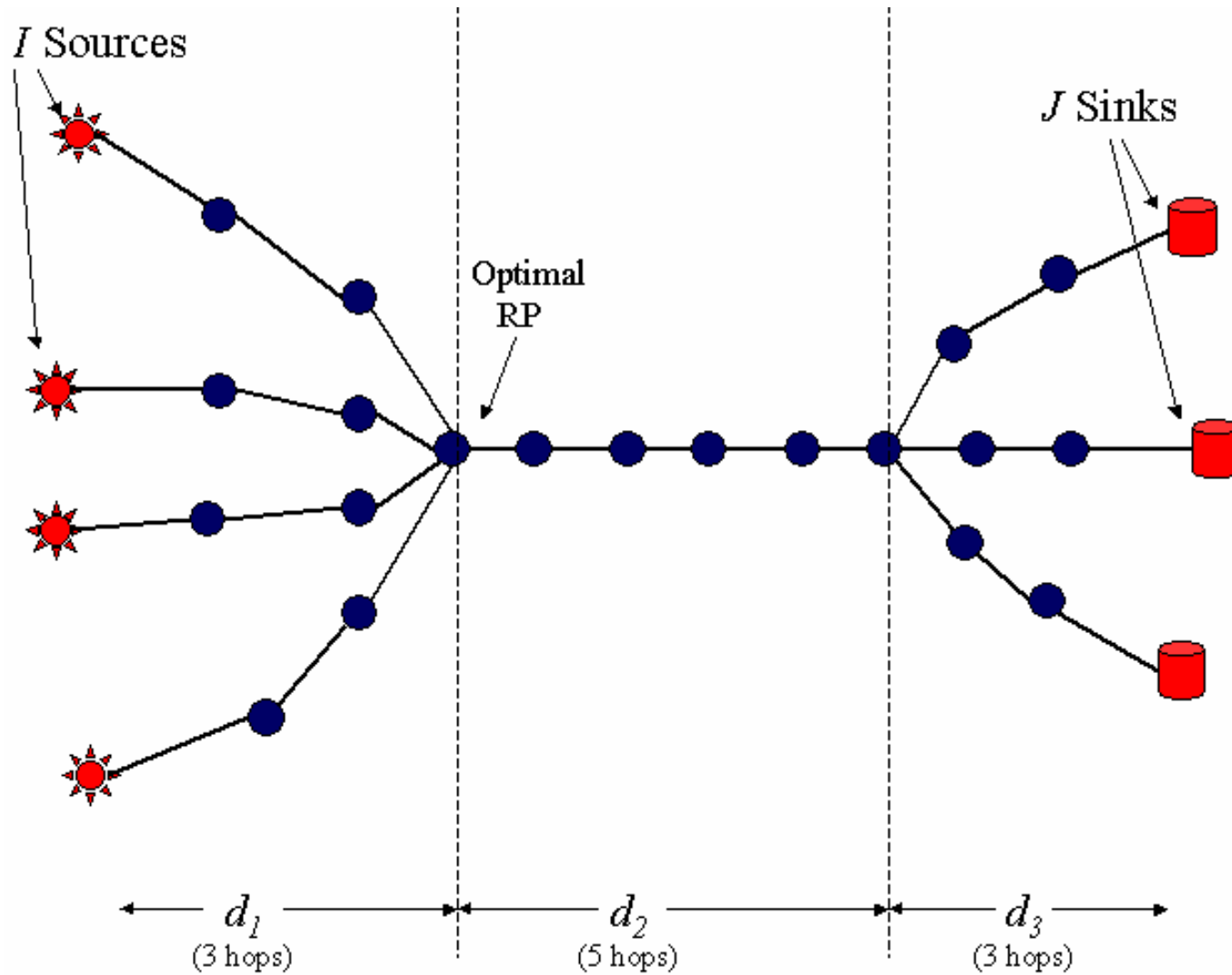
This brief taxonomy is from Reference [7] in the paper - Heidemann, Silva, Estrin.

# Workload Model



Sources are iid; sinks are iid

# Network Model



Network “fans-out” to both sources and sinks 6

# Additional Details of Model

- Topological parameters
  - $d_1, d_2, d_3, n, l, J$
- Traffic parameters ( $p_j, l_i$ )
- Data rate (and size) parameters
  - $S_I$  interest data rate per epoch
  - $S_R$  response data rate
  - $S_E$  exploratory data rate
  - $S_D$  application data rate
  - $\alpha$  fraction of application data sent with exploratory data
- Discrete time, expected case analysis

# Expanded Taxonomy of Diffusion Algorithms

- Authors analyze several variants of diffusion algorithms
- We look at 8 of these

	No Aggregation	Data Aggregation
<p><b>Flooding</b></p> <p>4</p>	<p><u>NAF</u></p> <ul style="list-style-type: none"> <li>• Pull (2-phase)</li> <li>• Pull (1-phase)</li> <li>• Push</li> </ul>	<p><u>AF</u></p> <ul style="list-style-type: none"> <li>• Pull (2-phase)</li> <li>• Pull (1-phase)</li> <li>• Push</li> </ul>
<p><b>Directed (Geographic)</b></p> <p>4</p>	<p><u>NAD</u></p> <ul style="list-style-type: none"> <li>• Pull (1-phase)</li> <li>• Push</li> </ul>	<p><u>AD</u></p> <ul style="list-style-type: none"> <li>• Pull (1-phase)</li> <li>• Push</li> </ul>
	<ul style="list-style-type: none"> <li>• Rendezvous</li> </ul>	<ul style="list-style-type: none"> <li>• Rendezvous</li> </ul>

# Example of Analysis (NAF Push)

- NAF  $\Rightarrow$  No aggregation of data, and flooding is used
- Push  $\Rightarrow$  Sensor data is pushed from sources to sinks
- Overhead =  $C_{push}^{NAF} / (U^{NAF} + C_{push}^{NAF})$

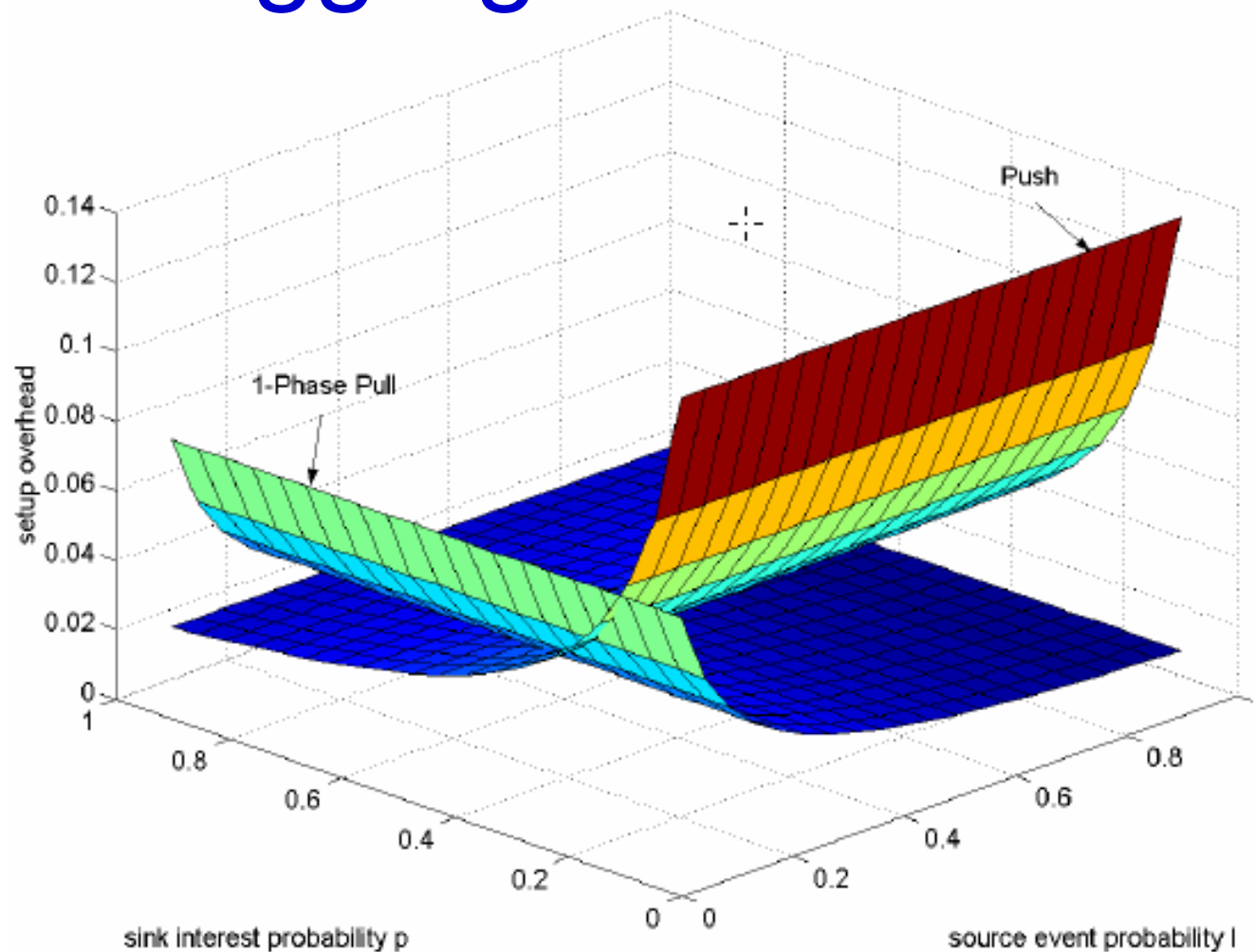
where

$$C_{push}^{NAF} = \sum_i l_i (S_{En} + \sum_j p_j (d_1 + d_2 + d_3) S_R) - \alpha U^{NAF} \quad (4)$$

and

$$U^{NAF} = \sum_i l_i \sum_j p_j S_D (d_1 + d_2 + d_3) \quad (1)$$

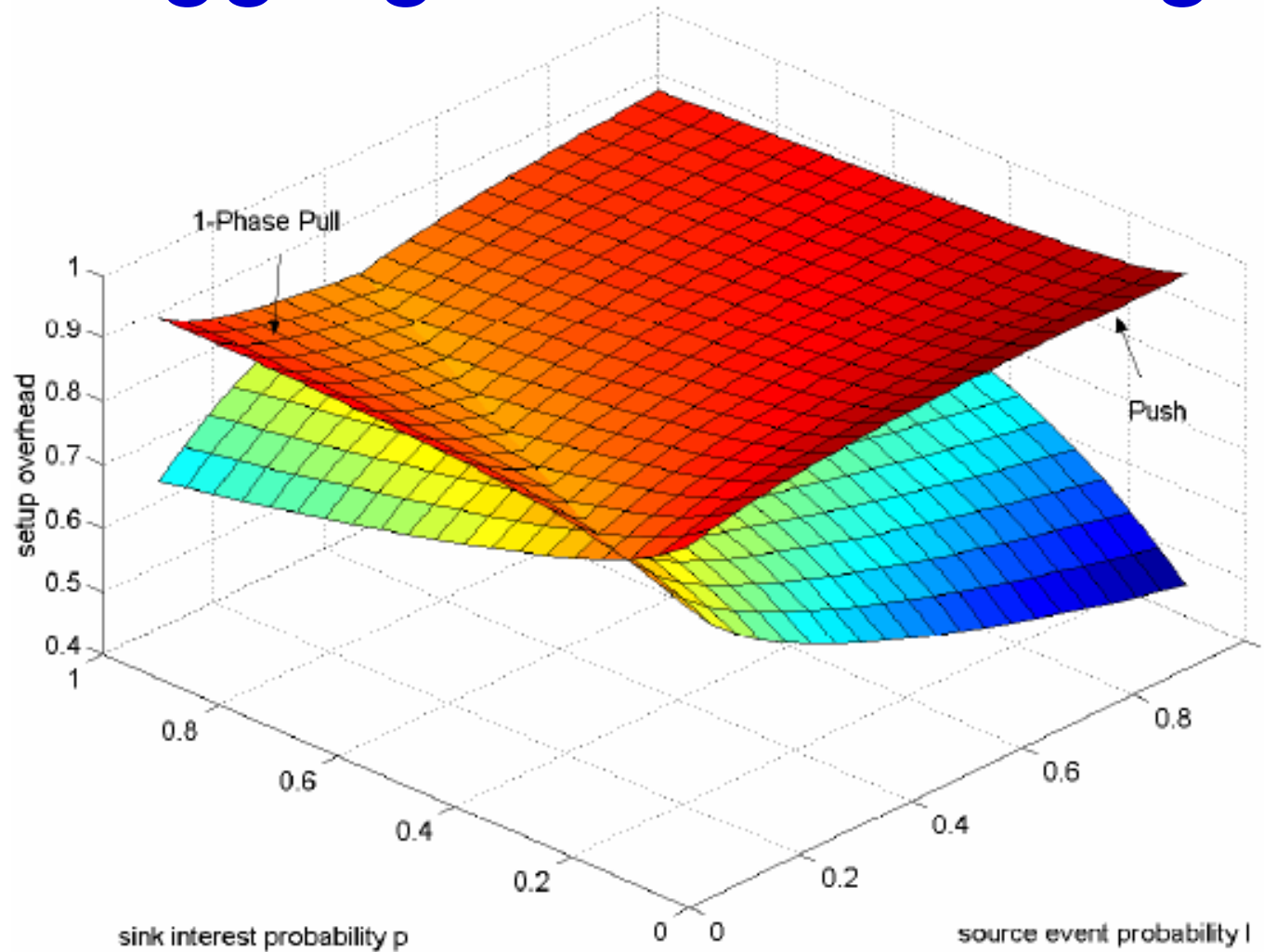
# No aggregation – Flooding



Push diffusion better for active sinks with quiet sources

Pull diffusion better for active sources with quiet sinks

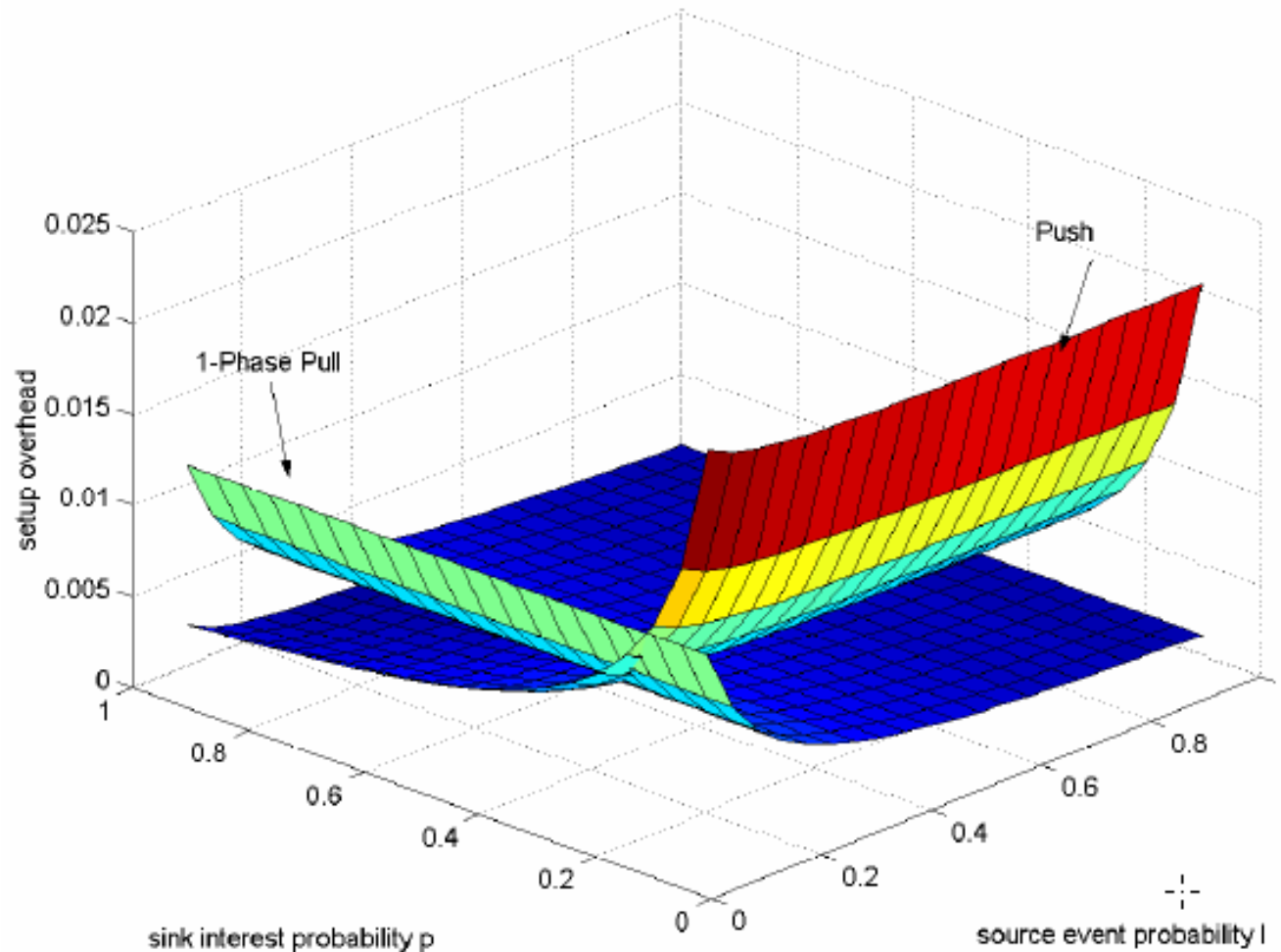
# Aggregation – Flooding



Aggregation significantly increases overall overhead

Aggregation “levels out” worst case behavior of push diffusion

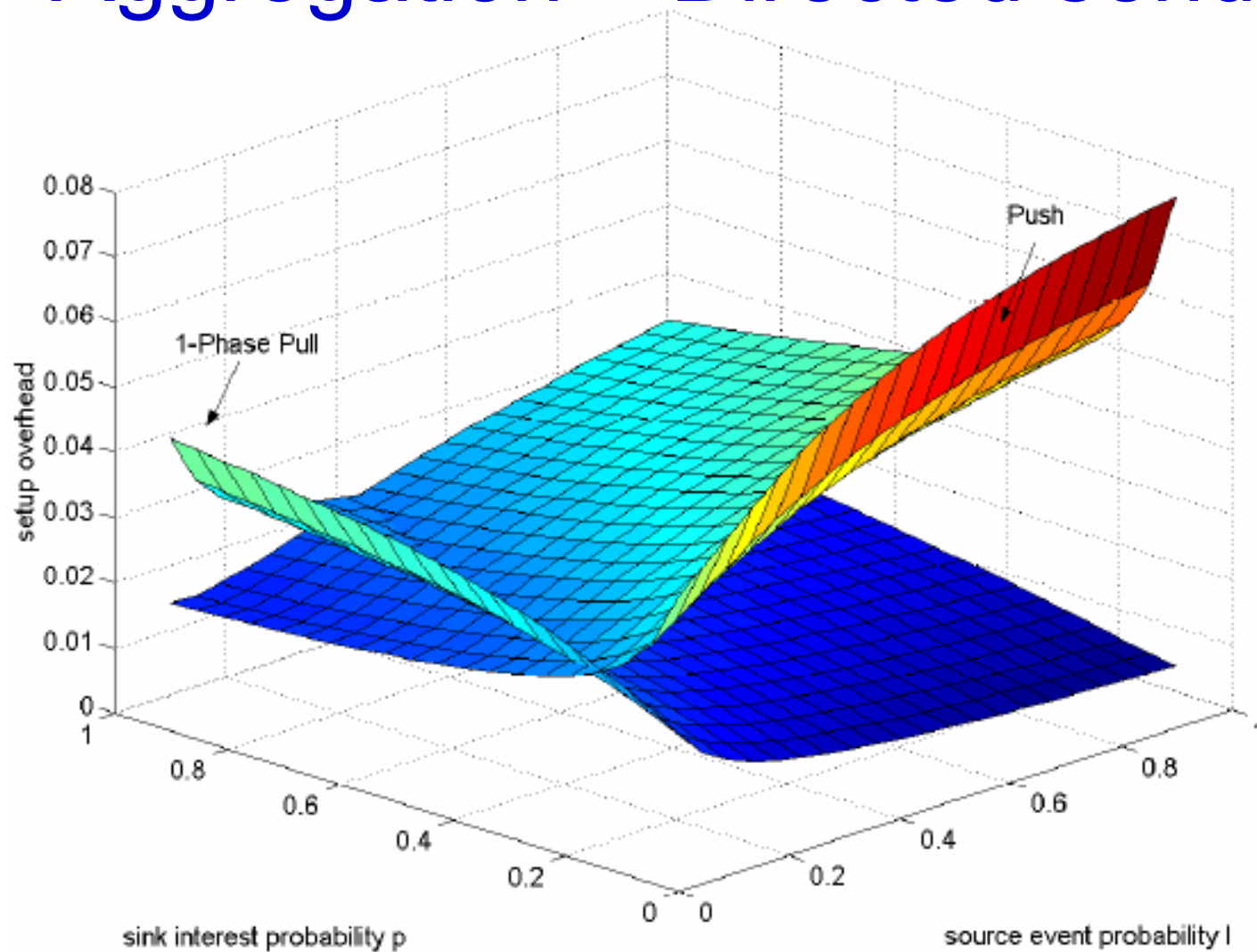
# No aggregation – Directed sends



Avoiding flooding significantly reduces overhead

Trends are the same as with no aggregation – with flooding

# Aggregation – Directed sends



Again, avoiding flooding significantly reduces overhead  
Trends are the same as with aggregation – with flooding<sup>13</sup>

# Application-specific Information Routing Summary

- Quantified when push diffusion outperforms pull diffusion (fewer active sources)
- Quantified when pull diffusion outperforms push diffusion (fewer active sinks)
- Mismatching application to routing method can yield performance penalty of 80% or more
- Rendezvous hybrid in theory can outperform both push and pull diffusion, but performance depends heavily on placement of rendezvous point

# What I liked (strengths)

- Nice piece of analysis:
  - Abstracts away application
  - Uses parameterized workload to capture application behavior
  - Tractable workload and network model
  - Good comparison of algorithms with respect to overhead
- Consistent with simulation results in companion paper:
  - Reference [7] in paper – Heidemann, Silva, Estrin.
- Design insights are interesting and helpful

# What I didn't like (weaknesses)

- Analysis includes some questionable assumptions
  - Packet sizes (which are application dependent)
  - Independent sources and independent sinks
  - Sources and sinks that are independent of each other
- We don't know how fixing of algorithm parameters impacts results
  - Packet sizes
  - Interest interval, exploratory interval, etc.
  - Others?
- One dimensional comparison
  - Overhead as fraction of control bytes vs. data bytes
- Inadequate investigation of rendezvous-style algorithms
  - Network topology not rich enough; one-dimensional comparison not enough

# Possible Future Work

- Different models for applications and sensor network
  - Heterogeneous applications querying sensor network
  - Heterogeneous environment stimulating data sources
  - Correlated workload sources
  - Richer network topologies
  - Data aggregation beyond duplicate suppression
  - Application traces for analysis and simulation (do they exist?)
- Within heterogeneous applications, what is a reasonable list of sensor types that we should be considering in deployments?
  - Are there good data / workload source models for these sensors?
- Within richer network topologies, most sensor networks are based on wireless communication in a 2-dimensional organization
  - What are the network topology properties that can be derived from this, and how do they impact the choice of information routing algorithm?

# Possible Future Work, cont'd

- Multidimensional comparison of algorithms, including “classic” Directed Diffusion
  - Comparison of *utility provided to applications*, not just *overhead*
  - What about side-effects of algorithms and impact on performance? For example, quality of gradient map, and ability to perform data aggregation?
- Experimental work to prove (or disprove) analysis and simulation with real-world applications
  - What is a reasonable set of representative applications?
- Avoiding flooding is clearly a good idea, but ... under what circumstances do applications exhibit spatial locality? And how do you program a sensor network to take advantage of this? This raises additional questions ...
  - What are good ideas for in-network processing to avoid flooding?
  - How could these enable directed / geographic routing?
  - What about nested queries, and aggregation at triggered sensors?
  - What about clustered computation, and aggregation at cluster heads?
  - Are there different kinds of spatial locality with different appropriate cluster organization and size?

# Possible Future Work, cont'd

- New diffusion algorithms for better application performance
  - Rendezvous-style algorithms (e.g., rendezvous nodes picked to work with your favorite routing algorithm – gossiping, GHT, GEAR, GPSR, etc.)
  - What are the advantages (if any) of 2-phase pull over 1-phase algorithms? What about 2-phase push where no data is sent with exploratory sends but is then pulled in phase 2?
  - Diffusion algorithms presented are parameterized (e.g., interest interval) –  
Could setting of these parameters be made adaptive?
  - Algorithms also encode transmission policy (i.e., rate-based vs. event-based) –  
Can these be made adaptive and / or hybridized?
- Others directions (From discussions in class)
  - The workload model on slide 5 seems unrealistic – not only are the iid assumptions questionable, but the fixed epoch time and the assumption that the dissemination algorithms are synchronized and well-behaved within each epoch might also be unrealistic.

# Possible Future Work, cont'd

- Within “different models for applications and sensor network” on slide 17, it would be interesting to investigate models for correlated sources and sinks that capture both the spatial and temporal locality that exists for both – perhaps even a two-state model for sources and sinks would be much better than the model used in this paper
- The analysis on slide 9 appears to present a loose upper bound on the overhead of NAF Push – Can this bound be tightened? What about the other analysis in the paper – Can it be made more accurate (or tighter)?
- Within “rendezvous-style algorithms” on slide 19, how would Zihui Ge’s work on publish / subscribe models be applicable to dissemination in sensor networks (see <http://www-net.cs.umass.edu/networks/publications.html> for references)?
- How would lessons learned from Distributed AI and Distributed MDP’s be applicable to information routing in sensor networks (see <http://dis.cs.umass.edu/pub/> for references)?
- How would lessons learned from Bayesian Networks be applicable to information routing in sensor networks (again, see <http://dis.cs.umass.edu/pub/> for references)? What are reasonable models for time and space in sensor networks for MDP’s and Bayesian Networks?