

Shh! Take Your Conversation Elsewhere

Exploiting Heterogeneity with Directed Diffusion

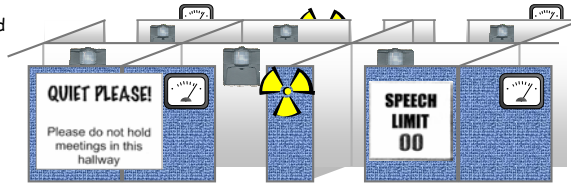
Omprakash Gnawali and Mark Yarvis

University of Southern California / Information Sciences Institute, Intel Corporation
<http://enl.usc.edu>

Motivation

Motion Sensors
 identify occupied cubicles

Warning Sign:
 a conventional weapon against noise



Sound-Level Sensors
 identify sources of noise

Alert Signal tells people to take their conversation elsewhere

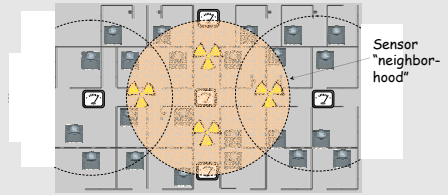
Goal: Promote productivity in a cubicle environment by reducing background noise

The "Do Not Disturb" Application

- Conventional Approach
 - Threatening signs
- Our Approach
 - Deploy sound-level and motion sensors
 - Actuate an alert when sound occurs near occupied cubicles
- Research focus
 - Leverage heterogeneity to reduce deployment costs and improve scaling and performance

Correlating Sensor Values

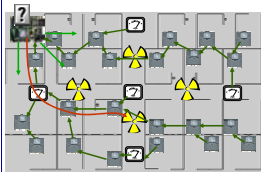
- Localize sounds that carry across a field of sound-level sensors
- Correlate sound propagation w/ motion sensors in a cubicle "neighborhood"



Exploiting Heterogeneity with Directed Diffusion

Implementation Using TinyDiffusion

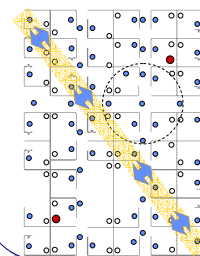
- Pull sensor data to a central location
 - A distributed approach requires too much data exchange



- Data processor sends an "interest"
 - Injected as flood
- Sensors return data values
 - Packets multihop to sink
- Server analyzes results
 - Is sound occurring near people?
- Server actuates alert signal

Processing Heterogeneity

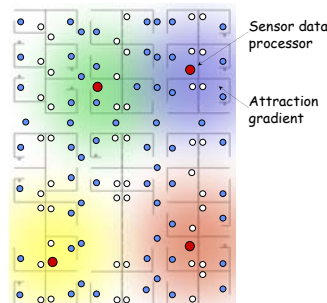
- Deploy two "high-end" nodes to *transparently* balance data fusion role



- Most sensors send data to one of two processing nodes
 - Use routing metric to select lowest cost path (i.e., most reliable or least energy cost)
- Some nodes send data to both processing nodes
 - Data from overlap area is required for both computations
- Transparency is key
 - System must work with any number of processing nodes

Scaling Up With Multiple Processors

- Larger networks require more processors
 - Ad hoc processor deployment
 - Each introduces attraction gradient
 - Sensors use metrics to choose the "best" processor
- Overlaps lead to non-linear scaling
 - More processors mean more overlaps
 - Many sensors must send duplicates of data to multiple processors
- How many processors for optimal scaling?



Acknowledgement: Ramesh Govindan and John Heidemann