Problem: How to delivery code-updates/new applications efficiently to a specified subset of nodes in a mobile wireless sensor network?

Scenario: Wildlife Monitoring

Sensors:
- Tmote Sky
- Meshbean

Challenges:
- Very sparse, mostly disconnected network
- Limited energy resources, thus very limited radio range and memory capacity
- Once the network is deployed, it is rarely possible to recollect the devices
- Requirements might change after initial results collected from deployment – reprogramming/new applications are needed!

Proposed solution: use the social characteristics of the network!
- Nodes (attached to animals) tend to form groups, thus meet some nodes more often then others – e.g. families -> Identify the groups/social links to delivery updates to specific nodes!
- Nodes tend to follow patterns in their movement – e.g. they go to feed to the same place
- Some nodes are more active then others – e.g. males go for food, while females/young animals stay around their “home-range” -> Identify more active nodes (“group leaders”) to delivery updates to specific groups!
- Not necessary to update all nodes – e.g. “update nodes often going to river X” or “update the node from each group, who stays around the nest most of the time” -> Characterise the nodes with attribute-value pairs (e.g. frequent-location= {nest, river}), and let the programmers decide which nodes to update using constraints based on the attributes

Initial Results of the dissemination protocol: (using ZebraNet traces, 20-node network)

Current direction: Selective update – provide the programming constructs to define constraints on the set of updated nodes, and combine this with the dissemination protocol

We used 30% less update transmissions than the epidemic-like protocol and achieved a similar delivery ratio!