

# Designing Protocols for Wireless Ad-Hoc Sensor Networks

Edo Biagioni

presenting work done in collaboration with

Kim Bridges and Brian Chee

and

Shu Chen, Wei Chen, Fengxian Fan,

Dan Morton, Ben Roy, Yihua Xie

November 2005

# Outline

- Introduction: wireless sensor networks
- Specific protocols:
  - Multipath On-Demand Routing
  - Lusus
  - SNDT
  - DiRT
  - GEO
- Summary



# Wireless Ad-Hoc networks

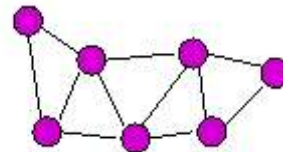
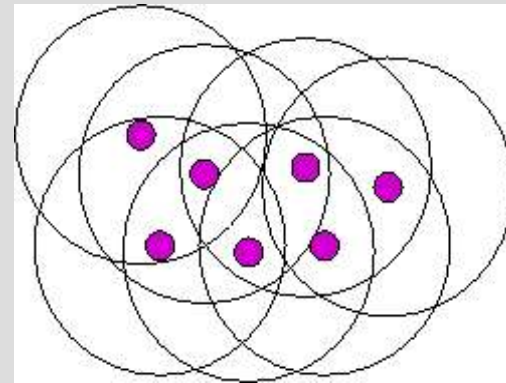
- each node:
  - has a radio transceiver
  - generates and receives data
  - transports data for other senders
- node distribution is usually random
- network discovery is dynamic
- main issues: where to send data (routing).  
how to send data efficiently
- could be mobile (MANet), fixed, or both

# Sensor Networks

- low-power sensors (battery powered?)
- deployed for long periods
- in remote locations
- data could be retrieved manually, but
- a network gives better latency (and requires less work :-)
- scalability is an issue

# Ad-hoc Wireless Sensor Networks

- monitor remote sites over extended periods at low cost: science, agriculture, tourism, military applications
- e.g. study endangered plants to find out why they are endangered
- e.g. monitor crops to determine when to water or fertilize
- environmental monitoring, and also images, and intrusion or herbivore detection

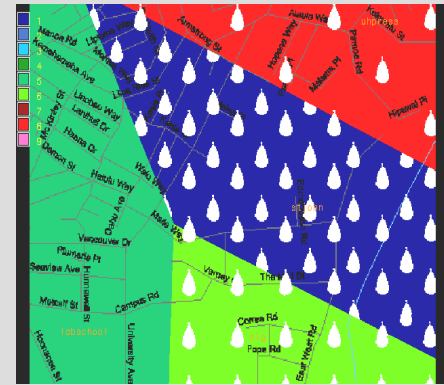


# Mobile Ad-Hoc Networks

- nodes assumed to be moving continuously and at random
- minimize routing overhead
- many protocols, including DSR, AODV
- few applications: UAVs, vehicles
- little study (so far) on general performance e.g. in transmitting TCP
- topology is not always changing: how do these protocols work in such cases?

# Challenges in Wireless Ad-Hoc Sensor Networks

- make sense of large amounts of data: visualization
- minimize the data transmitted: model generation, distributed event detection
- conserve power, e.g. by “sleeping”
- re-route around nodes that have failed and nodes that are congested, deal sanely with disconnection
- support encryption, heterogeneity
- The Capacity of Wireless Networks [Gupta and Kumar 1999] – can only send so much

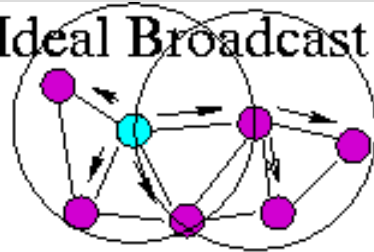


# Protocol Design Requirements

- like most networks, want reliability, high throughput, low delay
- low power requires:
  - low delay to completion: low packet delay and high throughput
  - high efficiency: do not transmit unnecessary packets
- synchronization needed so nodes can sleep
  - >> flooding broadcast always works

# Wireless Ad-Hoc Network Protocol Examples

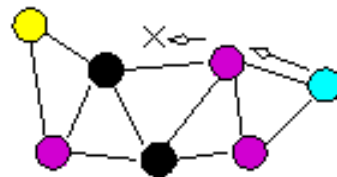
Ideal Broadcast



Flooding Broadcast



Synchronization needed  
for communication



-  source
-  destination
-  asleep

# WSN routing -- 2000

- mostly MANet protocols, e.g. DSR, AODV
- directed diffusion: communication paradigm, including broadcast of an *interest* which sets up a reverse-path gradient (hierarchical or tree routing)
- 802.11 supports peer-to-peer mode, but not (by itself) multi-hop
- can we improve on gradient?

# WSNs and MANets

- wireless sensor network nodes have even less power than MANet nodes
- nodes may be simpler than MANet nodes
- no motion (fixed – most common) or limited motion (fixed-mobile)
- it may be worth discovering good routes
- nodes may know position

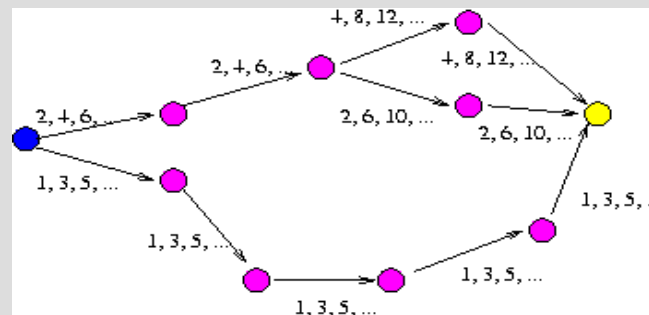
# building an environmental sensor network: PODS

- <http://www.pods.hawaii.edu>
- high resolution images
- sunlight, temperature, rain
- V0: wired sensor boards
- V1: PC-104 (PC-compatibles) running Linux, 802.11 for communications, BasicStamp power control board
- V2: Compaq Ipaq, Linux, 802.11
- V3: lower power ARM/Linux, not (yet?) done



# Multipath On-Demand Routing

- Shu Chen
- protocol to carry generic (IP) packets
- basic gradient routing: send a broadcast, reply along the reverse path
- may be multiple equal reverse paths: each node uses all in turn



# MOR reliability layer

- multiple next-hop nodes provide reliability, local load balancing
- if a route fails, does not (necessarily) require a new broadcast
- when a next-hop first fails, it is on probation
- deals well with local interference
- congestion: route restored on next attempt
- works well for occasional failures, not designed for continuous mobility

# MOR performance

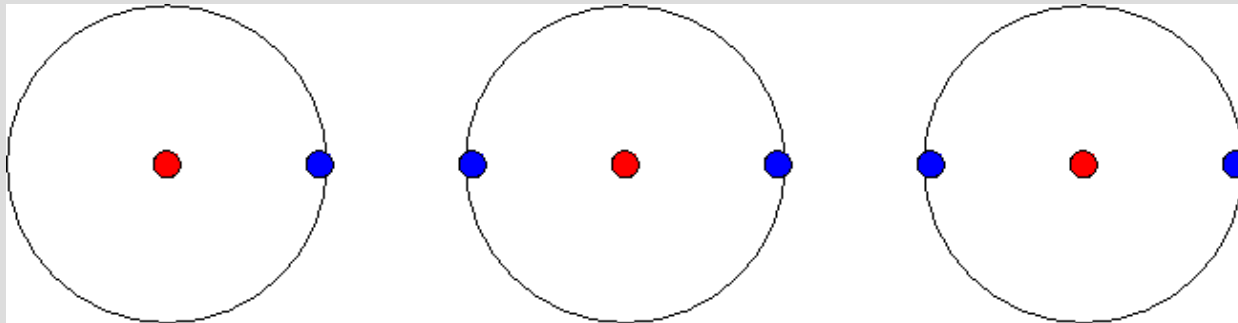
- faster transmission than DSR or AODV for given payload sent using TCP in large fixed ad-hoc wireless network
- should effectively route around congestion
- podr: long-term uninterrupted service in actual pods, also works under ns-2
- hop-by-hop ack (from 802.11 MAC) to decide whether to retransmit on this hop

# Measuring Performance

- MANet protocols usually tested in rather small (2-hop diameter) mobile networks
- how well do MANet protocols work when the network nodes do not move in the specific way of the one particular scenario?

# Typical challenge

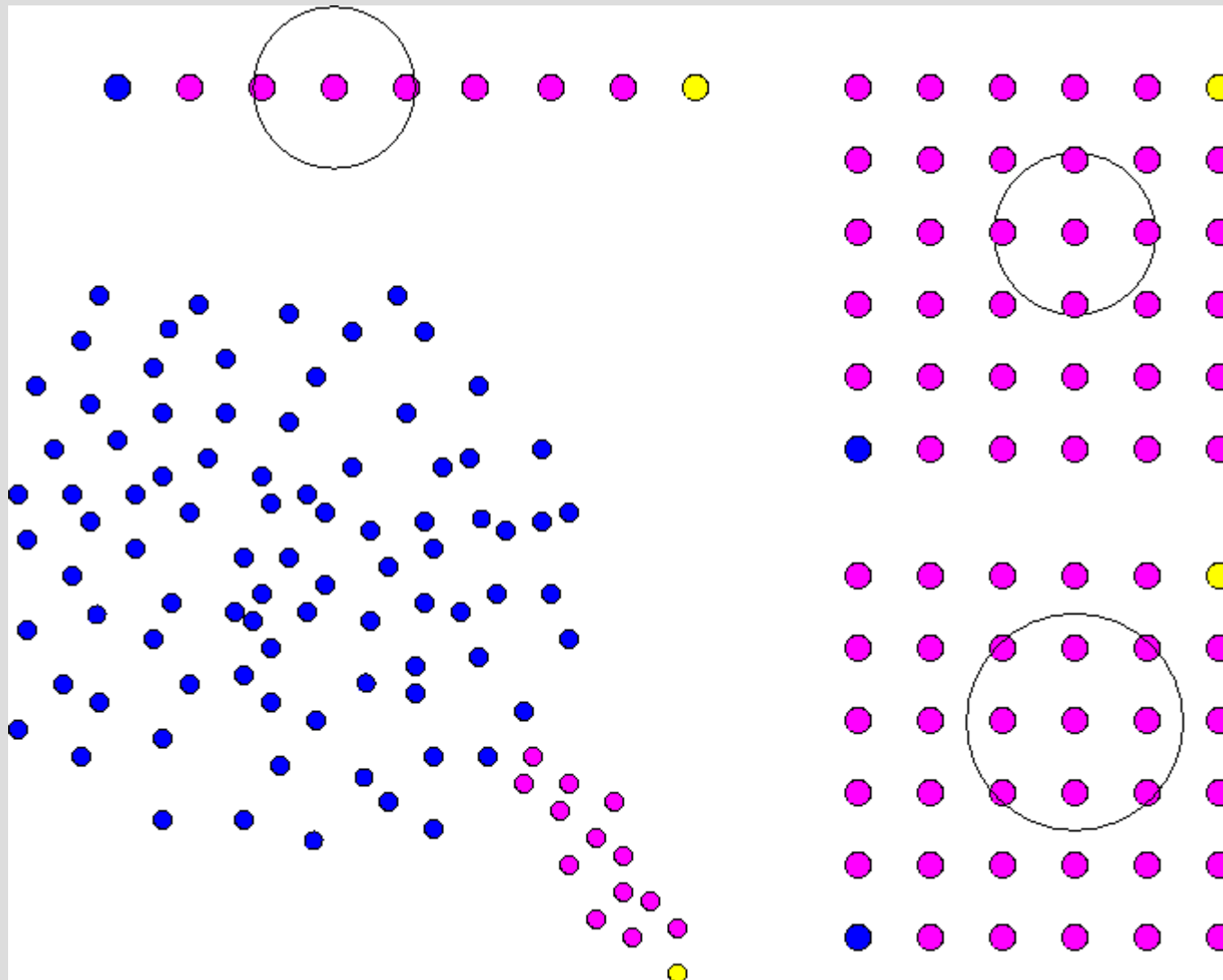
- co-ordinating transmission when there is only one path, and only 1-in-3 nodes can transmit at any given time



# MOR and test cases

- MOR works well in simple cases, e.g.
  - line: end-to-end transmission
  - 4-connected square: corner to opposite corner
  - 8-connected square
- MOR also works well in larger sensor-like networks:
  - 100 nodes distributed at random with long “tail” to base station
  - all-to-all random networks

# Specific MOR test cases



# Lusus Protocol

- Dan Morton
- really low power (500 $\mu$ W active, 5 $\mu$ W sleep)  
simple processors:  $\leq$  1KB RAM, 4K code
- Need a really simple protocol:
  - only send data (not IP)
  - only send to nearest base station
  - hop-by-hop ack
  - short packets, low overhead
- similar to diffusion, but simpler

# Lusus Techniques

- gradient: base stations regularly broadcast synchronization messages
- only send to nearest base station (base stations cannot be distinguished)
- data from different sources may be combined enroute
- messages are retransmitted if there is no ack from the next hop
- implementation tested on small network



# Distributed Route Table (DiRT)

- Ben Roy
- routing or forwarding to many hosts requires large routing tables
- distribute the routing tables so each node has at most  $O(\log(N))$  routes, yet every node can reach every other
- node IDs are set to be  $0..N-1$
- each node  $i$  has source routes for  $i\pm 1, i\pm 2, i\pm 4, i\pm 8, \text{etc.}$  --  $O(\log(N))$  routes

# DiRT routing

- for node  $i$ , if the destination  $D$  is in the routing table, send to it
- otherwise, for each destination  $j$  in the routing table, compute  $\Delta = D - j$ , the difference in addresses
- send to the destination with the smallest  $\Delta$
- this  $\Delta$  is always less than  $D - i$
- packets require at most  $O(\log(N))$  legs, each of maximum length the diameter of the network

# DiRT Open Issues

- Are there better ways to distribute large routing tables?
- for example, among neighbors
- if DiRT is used, what is the likelihood of shortcut routing?
- are we doing better than broadcast? and source routing tables could be  $O(N \log N)$
- how does network geometry affect performance?



# Geographic Routing

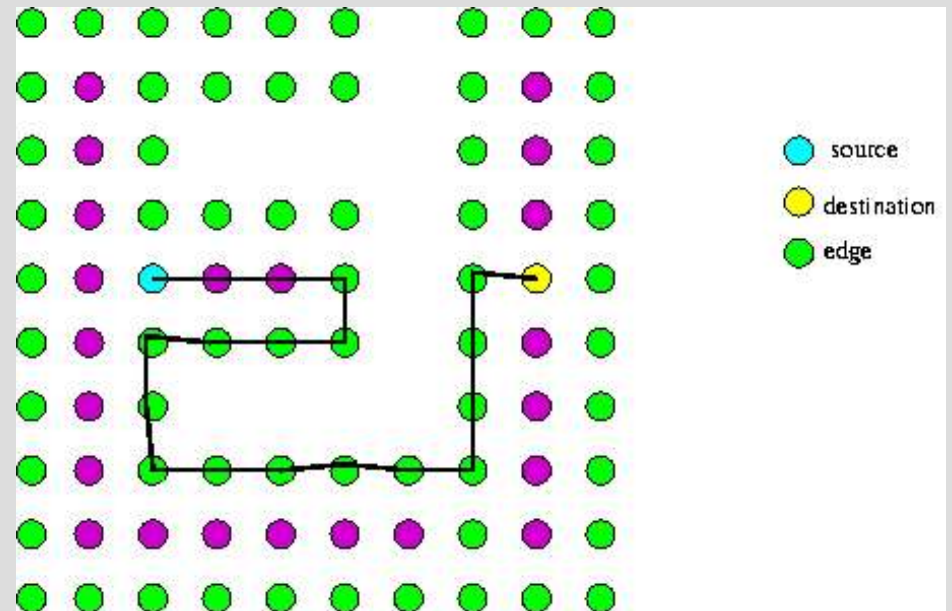
- each node knows its own position, e.g. via GPS
- if the destination is identified by position, simply route to the neighbor nearest the destination
- can cause routing loops, e.g. at dead ends
- many refinements possible (and published), but (imho) overall a lack of good solutions

# Geometric Routing: GEO

- geographic routing works fine as long as the network is densely connected
- only problems are at the edges of a connected area
- communicate the *geometry* of the connected areas, and route around any “holes”
- all the nodes on an *edge* must keep track of the geometry of **that** edge

# GEO implementation

- Wei Chen
- on the ns-2 simulator
- complex topologies simulated
- many nodes simulated
- scalability is good



# Related work: 2005

- Lots of protocols and ideas: Berkeley motes, diffusion, Zigbee, disjoint multipath, cluster-based schemes
- routing around low energy nodes
- SPINS: security protocols for sensor networks, by Perrig et al. (2001) --practical secure transmission on tiny processors
- sensor position and motion

# More related work: 2005

- Zigbee uses gradient most of the time, AODV for all-to-all communication
- still lots of interest in all aspects of sensor networks, e.g. HICSS 2006 minitrack on wireless sensor networks (co-chairs Anastasi, Biagioni, Olariu) has papers on: distributed processing, energy efficiency, and organization of wireless sensor networks

# PODS publications

- [www2.ics.hawaii.edu/~esb/pods/index.html](http://www2.ics.hawaii.edu/~esb/pods/index.html)
- A reliability layer for ad-hoc wireless sensor networks
- The application of remote sensor technology to assist the recovery of rare and endangered species
- An approach to data visualization and interpretation for sensor networks
- wireless sensor placement for reliable and efficient data collection

# Interesting Issues

- ad-hoc is new network technology:
  - IP assumptions don't work
  - connectivity may be intermittent
  - it is OK to design from scratch
- low power operation may be achieved at physical, data link, network, application layer, or through physical motion
- should data ever be unencrypted? How to do automatic encryption/authentication?

# More issues

- are there better ways of routing?
- are there reasonable “standard” benchmarks for routing? especially since routing can really matter
- integrating power aware routing, position determination, optimal node placement, architecture, operating system (e.g. TinyOS), packet scheduling, etc

# Summary

- Protocols to support goals of sensor network deployment: MOR, Lusus, SNTD, DiRT, GEO
- building actual sensor networks to evaluate and motivate the ideas



# Sensor Network Data Transmission: SNDT

- Lisa Fan and Yihua Xie
- IP over MOR, can use ssh, but:
- ssh has high overhead for small transfers
- want *connectionless* (but stateful) secure and efficient protocol
- use UDP, with explicit acks to minimize the overhead
- rate-limit transmission to avoid generating congestion

# SNDT strategies

- initial secret is part of node configuration
- initial secret used to exchange session keys
- nodes send data to base station
- base station sends commands and configuration to nodes
- authentication to prevent bad data, commands, configuration
- encryption to hide data, commands

# SNDT open issues

- synchronization must be done quickly, but authentication takes time
- even measuring RTT is hard if de/encryption is needed before ack
- each key must have an ID, which must be present on packets: does this make the attacker's life easier?
- maintaining session keys, and recovering from reboots, can be challenging

