Multi-hop Data Collection

Alec Woo, UCB
Terence Tong, UCB
Phil Buonadonna, Intel

Nest Summer Retreat 2003
June 18th, 2003
The Problem

- Design an ad hoc routing protocol for data collection
  - Create a many-to-few spanning tree topology
    - Focus on many-to-one first
  - Explore and characterize underlying connectivity options
  - Enhance end-to-end reliability by exploring quality/reliable routing paths to the base station
  - Maintain a stable routing tree
Three Core Components

- Connectivity Exploration
  - Build link reliability statistics of each neighbor through estimator

- Neighborhood Management
  - Maintain a subset of “good” neighbors using constant space (route table)
  - Cannot perform estimation unless neighbors are in the table

- Routing Protocol
Estimation

- Link estimator
  - Snoop to estimate reception success rate
- Time window average with EWMA smoothing
- Feedback estimations for bi-directional estimations

- Disadvantage:
  - Estimation latency can be high depending on message rate
    - 90% confidence of +/- 10% error takes ~100 samples
  - Received SNR can be a hint if radio supports it
Neighborhood Management

- Simple goodness criteria
  - Link reliability

- Frequency of packet reception as a hint to infer reliability
  - Rely on periodic messages or beaconing
  - Maintain frequently occurring neighbors
    - similar to page-table/cache management or
    - estimate the top-k frequent tokens in a data stream

- Frequency algorithm (Demaine et. al., VLDB 2002)
  - Table consistently maintains 50%-70% of its space for good neighbors when # neighbors > table size (up to 5 times in simulations.)
  - See paper for details
Routing Protocol

- Periodic route updates
  - Feedback link estimations to neighbors
  - Convey routing information

- Distance vector based protocol
  - Cost metrics
    - SP with threshold (filter out bad links)
    - Expected total transmissions = neighbor’s total transmissions + \( \frac{1}{\text{forward}} \times \frac{1}{\text{reverse}} \)

- Route damping
  - periodic parent evaluation rather than opportunistic based on route update arrivals
Parameters

Data rates

- Congestion => what’s the effect?

- Route update/Parent evaluation rates
  - Overhead => hinder bandwidth

- Transmission Power/Node Density
  - What’s the effect on transmission power settings?

- Upper bound on link retransmissions per hop

Queue management

- Each node is a source and a router
- Multiple bw. between forwarding and originating data
Evaluation!!

- Study it empirically by exploring the effect on different parameters
- Use naïve eviction policy to maintain route table

- Prototype routing stack in TinyOS
  - Mh6 version in April
    - Used by http://firebug.sourceforge.net/
  - Phil B. (Intel)
    - Merging with Surge implementation
Network Monitoring Tools

- Command Interpreter to vary settings
- Collect statistics on
  - Number of packets sent per node
  - Number of retransmissions per packet
  - Success rate delivered to base station
  - Cycle occurrence
  - Route stability
  - Hop over time
  - Forwarding queue status
  - Link qualities to parent
- Database archive
- Matlab
  - Interface to database for backend processing
  - Dynamically monitor the network status
Intel Test Bed

- Inside Intel Research Laboratory
  - 14 Ethernet modules with Mica nodes + 15 more scattered around Intel
  - Thanks Phil B. from Intel
  - Noisy indoor environment

- Experiments
  - transmission power is high (setting = 10)
  - Small scale 29 nodes
  - Rely on TinyOS synchronous Acks for retransmissions
    - Not too bad at high power setting
Success Rate (Intel)

- 29 Micas above ground
- power = 10
- 0.6 retrans. per packet trans. on average
- no cycles detected
- each exp runs >1hr
Tree Stability

- Stability at ~27% channel utilization
- Stability at ~50% channel utilization
Link Stability

- Stability of a link at ~27% channel utilization
- Stability of a link at ~50% channel utilization
Repeat the same experiment

- Larger scale 50 nodes
- Close to ground: 3 inches above
- Layout uniformly as a 5x10 grid
  - on ~4500 sq feet indoor setting
- Power setting is low (65)
  - Attempts to create more hops
- Grid distance equals 8 feet
- Limit to 3 retransmissions per hop
Grid Layout (Hearst Mining)
RFM Connectivity

- Recall connectivity graph for RFM
  - node falls within the BAD transitional region of RFM

- Network is actually not sparse!
  - Route table size is set to have 15 neighbors
  - We have a full table!!
  - Link qualities of neighbors varies from 80% to below 10%
What results do we get?

- Network partitions on routing
  - Hypothesis:
    - nodes (bs) with poor reliability is removed from route table
    - Eviction policy removes poor neighbors when table gets full
    - Link quality is so bad that routing avoids picking these links

- At low transmit power
  - Base station’s programming board attachment plays an effect on connectivity
  - Environment (such as wall/metal poles) plays a more important factor
    - Moving nodes away from walls/poles enhance connectivity
  - Easy to conclude => RFM doesn’t work
    - All it means is to place nodes in the “clear” region
    - => up the node density
What if you got it?

- 3 hop network
- Average is 81% including ~10% starting overhead
- ~1 retrans/packet trans.
- > 30% channel utilization
Other Issues

- Forwarding queues
  - Mostly empty at this particular uniform layout
  - Less importance on what queue management to use for multiplexing data and forwarding queue

- Robustness
  - Larger scale shows tricky bug
  - Task posting fails due to full task queues
Moving Forward

- Perform experiments to characterize ChipCon radio in different settings

- Not finish, there is more to understand!!!
  - Continue our evaluation plan to understand the effects of the different parameter settings
  - The two different radios too
  - Challenge: evaluation on low data rate, large scale networks take a long time!
    - Each single can take several hours

- Explore relationship of
  - Sleep scheduling vs. estimation vs. routing vs. neighborhood management