Flexible Power Scheduling

for multihop sensor networks

Barbara Hohlt <hohltb@cs.berkley.edu>

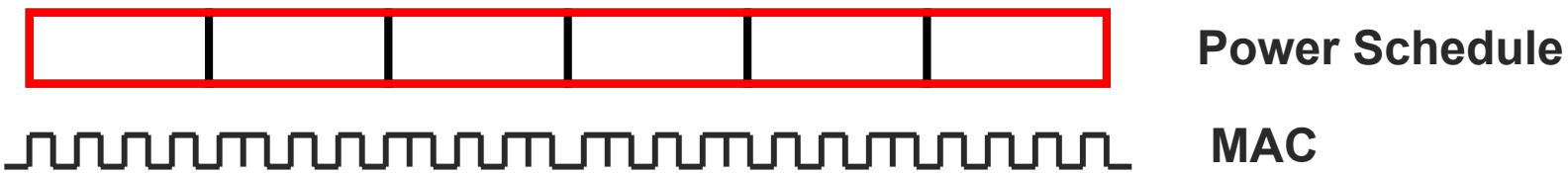
Eric Brewer <brewer@cs.berkeley.edu>

Nest Retreat

January 2004

FPS Two-Level Architecture

- Coarse-grain time division scheduling
  - Distributed local adaptive schedules
  - No global phases
  - Schedules Tx, Rx, and radio off times
- Fine-grain CSMA MAC underneath
  - Collision avoidance, slot sharing



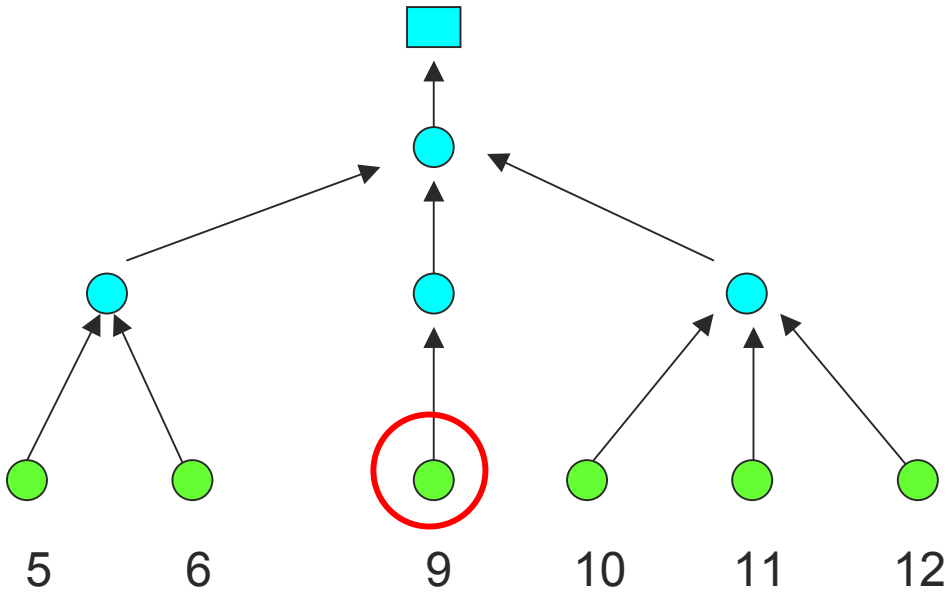
Benefits of Slotted Schedules

- Power savings:
  it is clear when to turn the radio off
- Fair bandwidth allocation
- Reserved bandwidth for the flow
- No traffic correlation effect

Tests

- Tests
  - 10 MICA1 motes plus base station
  - 6 motes send 100 messages across 3 hops
  - One message per cycle (3200ms)
  - Begin with injected start message
  - Repeat 11 times
- Software
  - FPS - 80ms slots, 40 spc, power management off
  - Naïve - simple multihop, store and forward

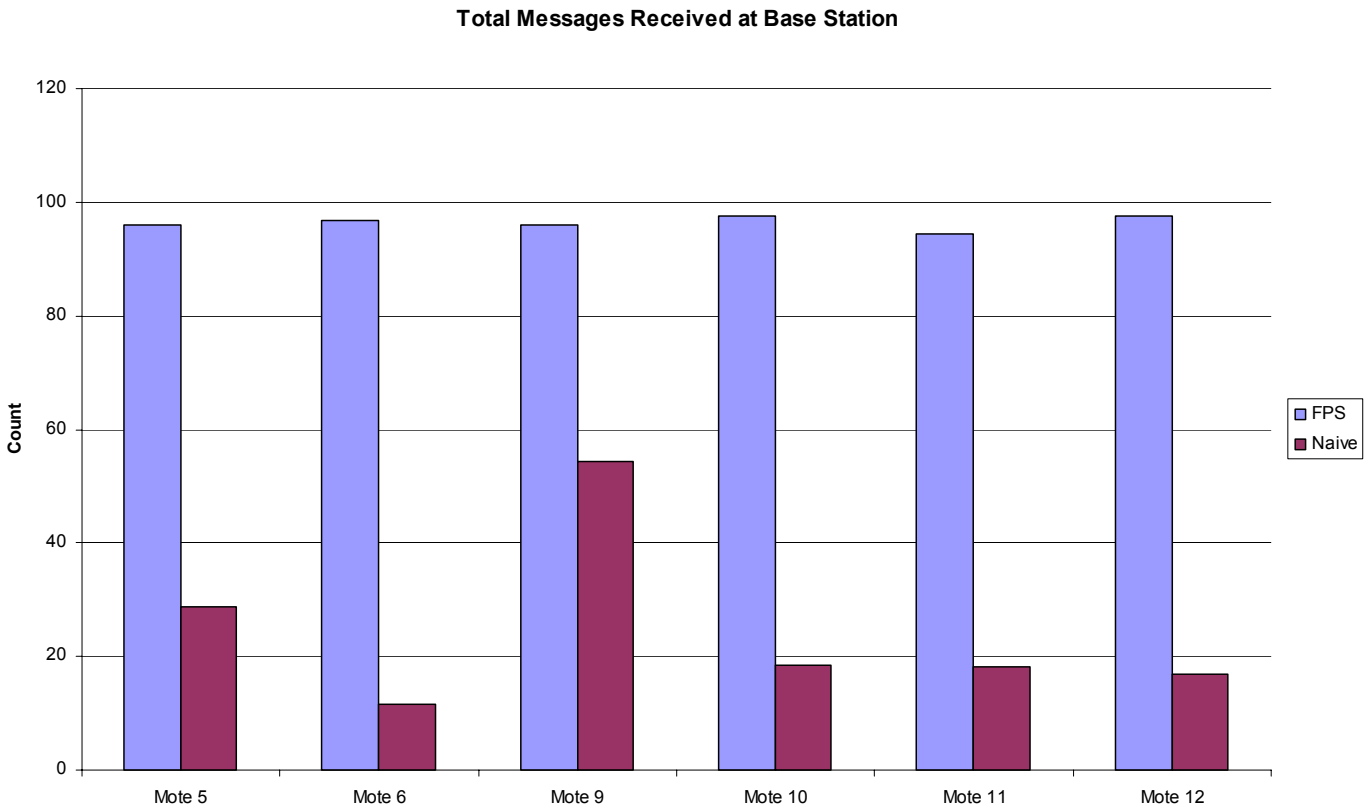
Two Test Topologies



- Dinnertable
  - one area
  - 8’ x 3’4”
- Household
  - five areas
  - motes are 9’-22’ apart

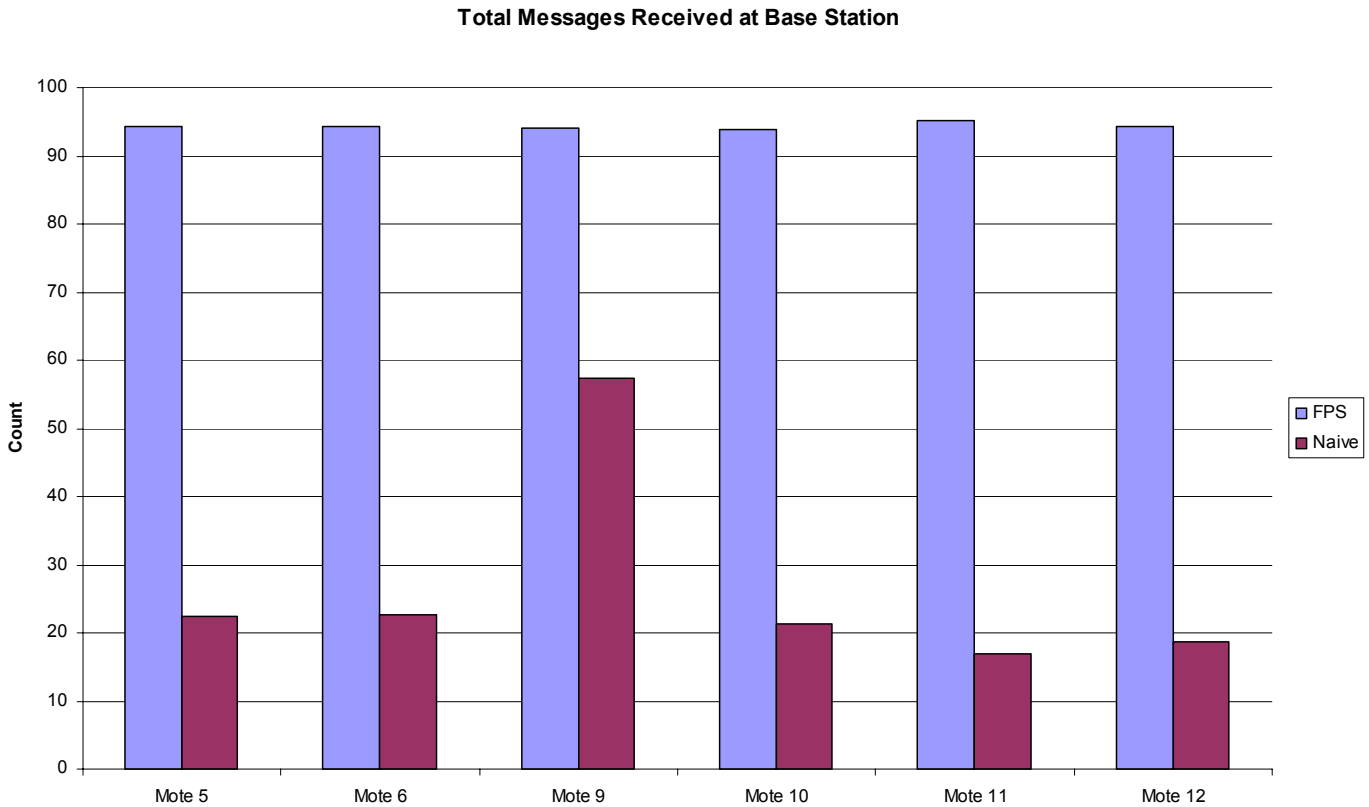
*Note: In FPS tests, Mote 9 is 3 hops from the base station.
In Naïve tests, Mote 9 is 2 hops from the base station.

Household Yield Tests



*Note: In FPS tests, Mote 9 is 3 hops from the base station.
In Naïve tests, Mote 9 is 2 hops from the base station.

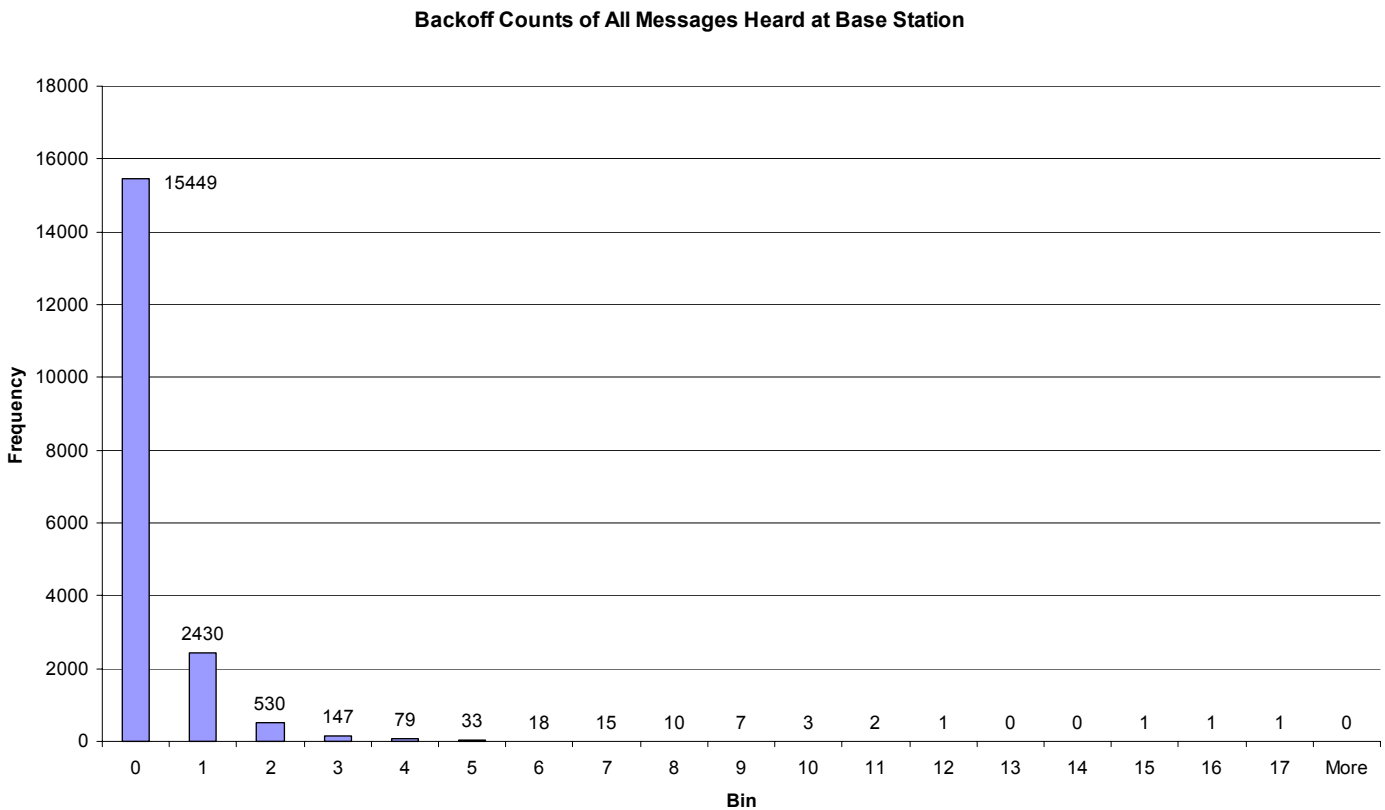
Dinnertable Yield Tests



*Note: In FPS tests, Mote 9 is 3 hops from the base station.
In Naïve tests, Mote 9 is 2 hops from the base station.

Dinnertable Backoff Histogram

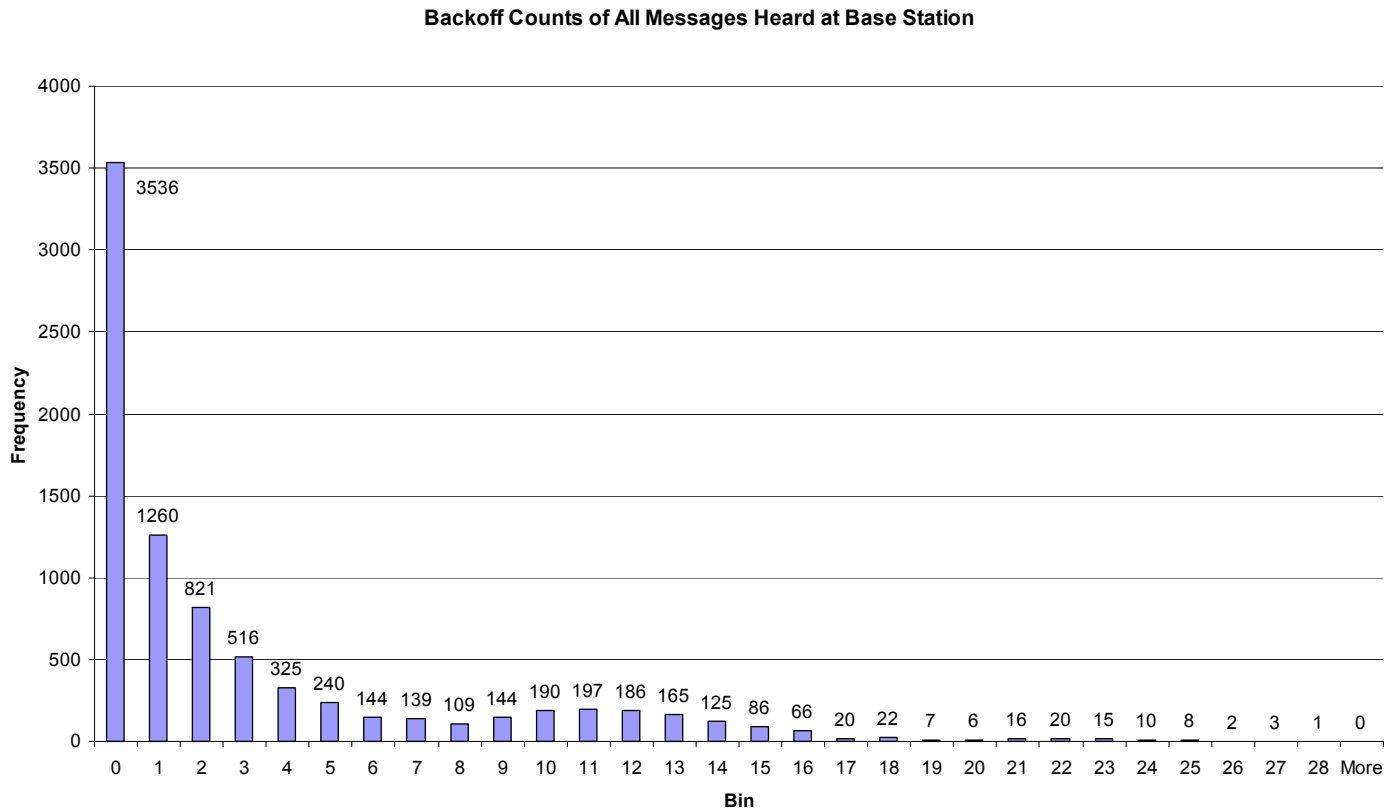
FPS



18727 of 19800 Messages Heard Across All Hops

Dinnertable Backoff Histogram

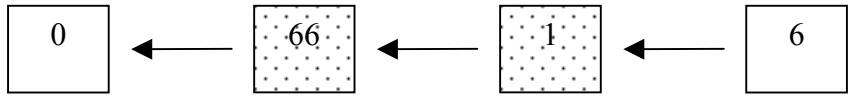
Naive



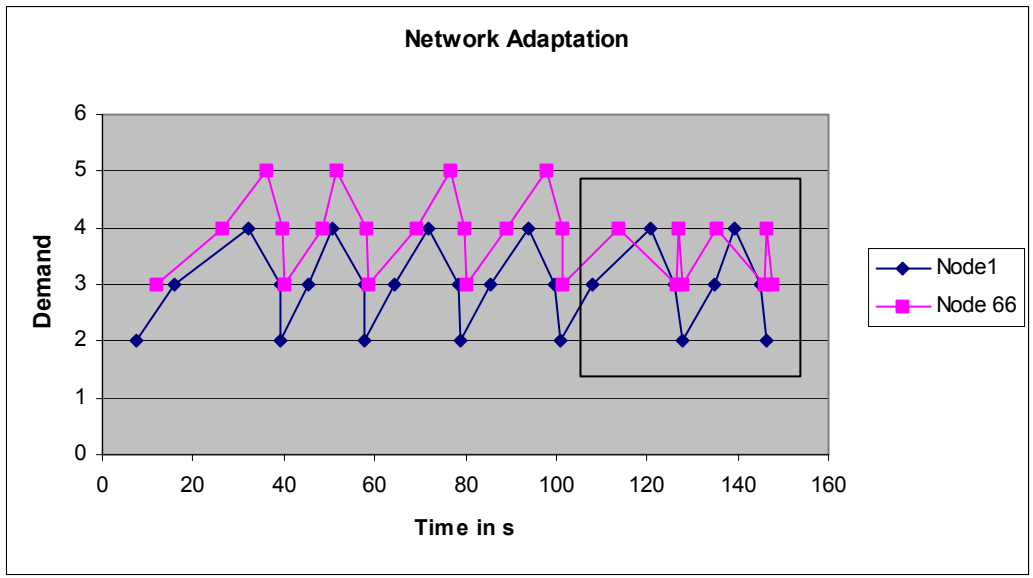
8379 of 19800 Messages Heard Across All Hops

END

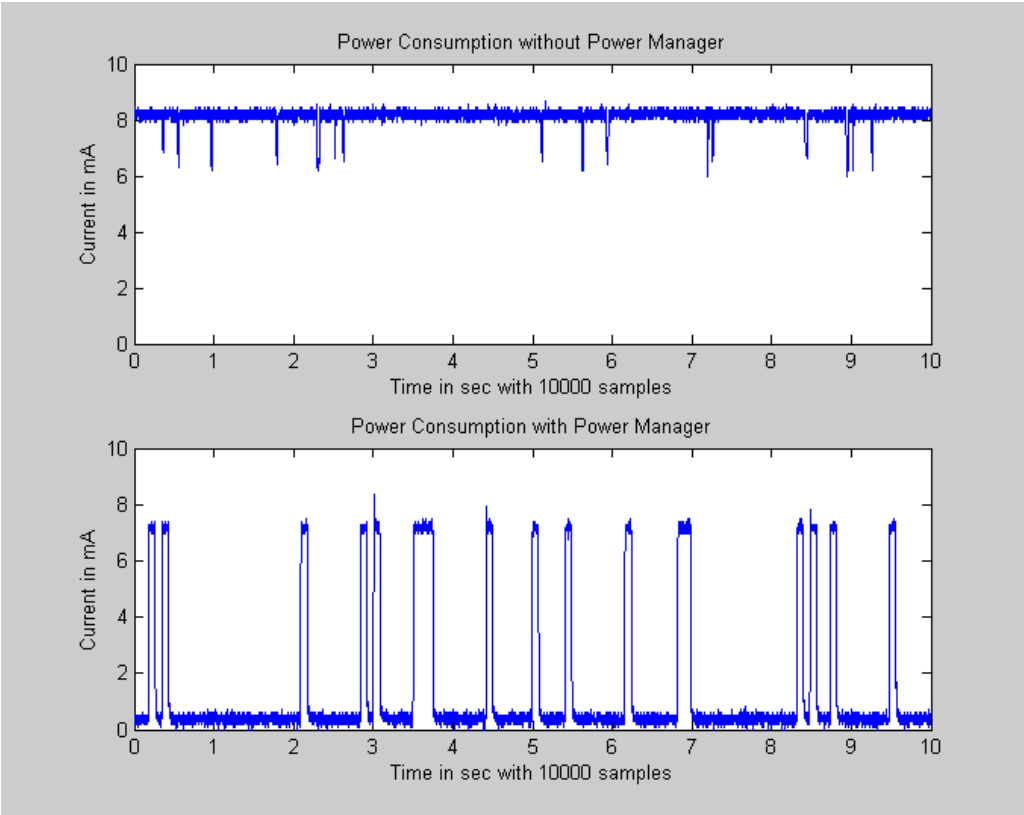
Network Adaptation



Node 1 and Node 66 adapt to fluctuating demand in the network as Node 6 increases and decreases its demand by 2.



Power Consumption



The measured current at an intermediate node over 10 seconds. A data packet is received and forwarded once per cycle, every 2.6 seconds.

Principles of Power Scheduling

- Need to know when to listen (whole stack)
- Schedule flows not packets
- Adaptive
- Two-levels
- Nodes that want to change listen
- Requires global and local views

END