PEDAMACS: Power efficient and delay aware medium access protocol for sensor networks

Sinem Coleri and Pravin Varaiya
Department of Electrical Engineering and Computer Science
University of California, Berkeley
Outline

- Introduction
- Special class of sensor networks for traffic applications
- Pedamacs networks vs random access networks
- Pedamacs protocol
- Comparison with random access networks
- Suitability for traffic applications
Introduction

- Sensor networks are designed for particular applications unlike general purpose wireless networks

- Attraction is wireless, low-cost—easy to deploy

- Limiting factor is power consumption; biggest power consumer is radio—importance of MAC protocol

- Biggest cost will be deployment and maintenance—lifetime will determine economic feasibility

- Work needed in signal processing, reliability, failures
Sensor networks for traffic

- Nodes generate data periodically (e.g., 30 s) for transmission to access point (AP) that is connected to ‘outside world’

- Nodes are power- and energy-limited; access points are not
Current traffic sensing technology

- Loop detectors is the standard; loops last 10 years
- Closing lane to cut loops in freeway pavement is very disruptive
- Alternatives include: microwave side fired radar, video cameras
- Cost of these systems is $600-$1000 per detector (lane) per year
- Can sensor networks compete?
Sensor networks with two special characteristics

1. One distinguished node, Access Point or AP; sensor nodes or SN periodically (e.g., 30 s) generate data for transmission to access point.

2. SNs are power- and energy-limited but AP is not. Consequently

Transmission AP → SN is one-hop
Transmission SN → AP is multi-hop

- Two conditions satisfied in traffic applications

![Diagram of sensor networks with two special characteristics]
Performance criteria

- **Power efficiency**: radio is the biggest consumer; power wasted in collisions, idle listening (when no packets are received), overhearing unintended packets, transmission of control packets.

- **Real-time guarantee**: needed for process control applications as in case of traffic control; network may not be used without such guarantee.

- **Congestion control**: nodes near AP must transmit more data (from upstream nodes).

- **Fairness**: each node must transmit its data every period.
Prototype Berkeley sensor node

- Assembled from off-the-shelf components (retail cost $100)
- 4Mhz, 8bit MCU (ATMEL)
  - 512 bytes RAM, 8K ROM
- 900Mhz Radio (RF Monolithics)
  - 10-100 ft. range
- Magnetic Sensor & Acoustic Sensor
- LED outputs
- Serial Port

1.5” x 1.5”
## Power consumption of Berkeley node

<table>
<thead>
<tr>
<th>Operation</th>
<th>Power consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitting one 37B packet</td>
<td>0.92 mJ</td>
</tr>
<tr>
<td>Receiving one packet</td>
<td>0.60 mJ</td>
</tr>
<tr>
<td>Listening to channel</td>
<td>20.71 mJ/sec</td>
</tr>
<tr>
<td>Radio in sleep mode</td>
<td>15 μJ/sec</td>
</tr>
<tr>
<td>Clocking energy</td>
<td>294 μJ/sec</td>
</tr>
<tr>
<td>Sampling sensor</td>
<td>1.5 μJ/sec</td>
</tr>
</tbody>
</table>

2 AA batteries provide 2200 mAh or 0.25 mAyear @ 3V
Radio waste

- Collision
- Overhearing
- Idle listening
- Transmission of control packets
- Congestion control
Reducing radio waste—PHY and MAC

- Random back-off to reduce collision, intuitive congestion control
- 2 radios, one for wakeup signals, other for data—avoids overhearing
- RTS-CTS avoids collision, approximate TDMA, requires synchronization
- Use different channels or codes to reduce collision
- Use short control packets
- Save between 10-70 percent power over random access

Pedamacs vs random access networks

**Pedamacs networks:**
- Access point discovers network topology; nodes discover next hop
- Access point computes and broadcasts transmission schedule to all nodes (TDMA data)
- During data phase, node sleeps if it is not scheduled to listen or to transmit

**Random access networks:**
- Access point and nodes discover next hop
- Nodes randomly transmit and constantly listen for incoming packets
- Refinements proposed to reduce node ‘listening’ time
50 kbps; one packet every 30 sec; vertical scale is $\log_{10}$

- Listening in random access uses $1000X$ more power, and receiving uses $10X$ power than in Pedamacs
Two AA batteries: 2200 mA at 3 V

Pedamacs network lasts 600 days, need 5X improvement

Random access network lasts 10 days—unsuited for traffic control
**Pedamacs protocol**

- Assumes 3 transmission ranges:
  - **Longest range** reaches all nodes from AP
  - AP and nodes have **medium range** to identify neighbors, interferers
  - Nodes have **minimum range** (that guarantees connectivity) for data transmission

- Protocol phases
  - Topology learning: Each node identifies neighbor, interferers, and parent in AP-rooted tree
  - Topology collection: Each node sends this local topology information to AP
  - Scheduling and transmission: AP calculates and broadcasts schedule. Each node transmits and listens at predetermined slots; sleeps in other slots
Protocol packet structure

(a) general packet

2B 2B 1B 30B 2B
source address destination address message handler ID data CRC

(b) topology learning and topology collection coordination packets

8B 8B
current time next time -

(c) tree construction packet

1B
no of hops -

(d) local topology packet

1B 1B 1B 1B 1B 1B
node ID node level node parent no of nbrs nbr1 ... no of intfrs intf1 ...

(e) scheduling packet

16B 1B 1B 1B 1B 1B
current, next time slot seq no no of nodes node ID node ID no of nodes node ID ...
## Protocol phases

<table>
<thead>
<tr>
<th>AP</th>
<th>Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>‣ Broadcasts topology learning pkt, w. current time, next time</td>
<td>‣ Synchronize w. current time, record next time</td>
</tr>
<tr>
<td>‣ Starts flooding w. tree construction pkt</td>
<td>‣ Increment hop count, identify parent, neighbors, interferers (signal &gt; or &lt; threshold)</td>
</tr>
<tr>
<td>‣ Broadcasts topology collection pkt at next time</td>
<td>‣ Synchronize w. current time, record next time, transmit local topology pkt</td>
</tr>
<tr>
<td>‣ Learns topology. Computes schedule; broadcasts scheduling pkt at next time</td>
<td>‣ Synchronize w. current time, record next time. Transmit/receive scheduled data pkt</td>
</tr>
<tr>
<td>‣ Broadcasts topology learning pkt (if topology is outdated) or broadcast scheduling pkt</td>
<td></td>
</tr>
</tbody>
</table>
Protocol phases

If topology errors:
- AP: Broadcast topology learning pkt
- Nodes: Synchronize w. current time, record next time
  - Increment hop count, identify parent, nbrs, interferers
  - Broadcast topology learning pkt
  - CSMA
  - Synchronize, next time, transmit local topo. pkt
  - CSMA + ACK
  - Transmit/receive data pkt on schedule

If no topo errors:
- Compute, broadcast scheduling pkt
**Topology learning**

1. Current time
2. Next time

- AP broadcasts topology learning coordination packet to synchronize nodes and declare next time.
- AP sends the tree construction packet using medium range transmission.
- Tree construction packet flooded by sensor nodes using CSMA scheme.
- Sensor nodes determine neighbors and interferers from received tree construction packets.
**Topology collection**

AP broadcasts topology collection coordination packet to synchronize nodes and declare next time.

Each node transmits its local topology packet, listing its neighbors and interferers using CSMA scheme with implicit acknowledgement.

AP has complete topology information.
Scheduling phase

- Divided into time slots. Each node when to listen and when to transmit. Time slot carries one data packet.

<table>
<thead>
<tr>
<th>Slot number</th>
<th>Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>s2,s3</td>
</tr>
<tr>
<td>2</td>
<td>s1</td>
</tr>
<tr>
<td>3</td>
<td>s1</td>
</tr>
</tbody>
</table>
Pedamacs scheduling problem (AP)

- At end of topology collection phase,
  - tree \( G = (V, E) \), rooted at AP
  - interference graph \( C = (V, I) : (u, v) \in I \) if \( u, v \) interfere

- Construct conflict graph \( GC = (V, EC) \)

  \[(i, j) \in EC \text{ if } \begin{cases} (i, j) \in E, \text{ parent, child should not transmit} \\ (i, c_j) \in I, [(i, j) \in I] \wedge [c_j \text{ child of } j] \wedge (i, c_j) \in EC \\ i, j \text{ have same parent} \\ i, j \text{ are parent, child of same node} \end{cases} \]

- Neighbors in conflict graph should not transmit at same time

- \( E \) is collection of links to be scheduled
Example of conflict graph

\[ G = (V, E, I) \]

\[ GC = (V, EC) \]
Scheduling problem

- **Scheduling problem**: Given interference graph, find minimum length frame during which each node can transmit one data packet.

Theorem: scheduling problem is NP complete
Proof: reduce chromatic number problem to scheduling problem

Note: chromatic number problem is to find smallest $k$ such that graph is $k$ colorable.

Theorem: min schedule is at least $|V| - 1$ and at most $\frac{1}{2}|V||V| - 1$)
Proof. One packet received per step. Worst case is complete, linear graph.
Pedamacs schedule: obtain linear network

1. Obtain linear network $GL = (VL, EL)$ from tree $G = (V, E)$
   - $VL = \{AP, v_1, \ldots, v_N\}, v_n$ corresponds to all nodes at level $n$
   - $CL = (VL, IL), (v_i, v_j) \in IL$, if nodes at levels $i, j$ interfere
Pedamacs schedule: color linear network

1. Color linear network. Assign colors to each node.

2. Add colors so that nodes with same color form maximal non-conflicting set

Theorem: Linear network coloring takes $O(|V|^2M)$ steps. If max level interference in $GL = (VL, EL)$ is $K, M = K + 2$
Example

$GL=(VL,EL,IL)$

<table>
<thead>
<tr>
<th>node ID</th>
<th>colors assigned in the first phase</th>
<th>colors assigned in the second phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>{c1}</td>
<td>{c1, c4}</td>
</tr>
<tr>
<td>v2</td>
<td>{c2}</td>
<td>{c2}</td>
</tr>
<tr>
<td>v3</td>
<td>{c3}</td>
<td>{c3}</td>
</tr>
<tr>
<td>v4</td>
<td>{c1}</td>
<td>{c1}</td>
</tr>
<tr>
<td>v5</td>
<td>{c4}</td>
<td>{c4}</td>
</tr>
<tr>
<td>v6</td>
<td>{c2}</td>
<td>{c2, c3}</td>
</tr>
</tbody>
</table>
Pedamacs schedule: color original network

3. If nodes in linear network have same color, one node in original network at each corresponding level can transmit at the same time
   - Create superslot of consecutive slots. In each superslot, there is transmission of one packet from each level.
   - Size of superslot = number of colors
   - Number of superslots is at most number of nodes
Example

<table>
<thead>
<tr>
<th>Slot number</th>
<th>Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(yellow)</td>
<td>s1</td>
</tr>
<tr>
<td>2(blue)</td>
<td>s3</td>
</tr>
<tr>
<td>3(yellow)</td>
<td>s1</td>
</tr>
<tr>
<td>4(yellow)</td>
<td>s2</td>
</tr>
</tbody>
</table>

Note: s2, s3 could have transmitted in same slot!
**Special cases**

Max frame length \((K + 2)(|V| - 1)\)  Max frame length is \(3(|V| - 1)\)

\[
|i - j| \leq K
\]

\(v_i\)

\(v_j\)

\(v_N\)

**Diagram:**

- Left side:
  - AP
  - \(v_i\)
  - \(v_j\)
  - \(v_N\)
  - \(|i - j| \leq K\)

- Right side:
  - AP
  - \(v_N\)
  - Superslot

30
Comparison of random access and Pedamacs networks

- Comparison via TOSSIM, a TinyOS simulator

- Need to select critical parameters for comparison
  - Backoff-listening random access scheme
    - Back-off window, listening window
  - Transmission range

- Nodes randomly distributed inside unit circle
Simulation

- Pedamacs channel access mechanisms
  - without backoff-listening (AP)
  - with back-off listening (topology learning)
  - with back-off listening + implicit ACK (topology collection)
  - TDMA (nodes in scheduling phase)

- Random access channel mechanisms
  - data packets from nodes to AP use back-off listening + implicit ACK
  - flooding for topology discovery using back-off listening
Transmission range selection

- Minimum range that leaves network connected
Backoff & listening window selection for learning

- Percent of nodes reached by flooding from AP (topology learning): all nodes reached with large backoff window
Backoff & listening window selection for learning

- Max delay (bit times) experienced by tree construction packet during flooding depends on backoff window.
Backoff & listening window selection for collection

- Percent nodes whose packets reach AP with ACK, w/o Ack
Pedamacs parameters

- Percent successfully scheduled nodes

![Graph showing percentage of nodes vs. listening window size](image-url)
Selecting ack window sizes for 30 nodes

- Back-off window size 32768, listening window size 1024 bit times
Pedamacs vs random access delay

- Random access delay excessive for traffic application
Lifetime of Pedamacs network vs packet period
Improvements in lifetime

- Pedamacs networks can be made redundant by placing multiple nodes in cluster-only one node is activated.

- Additional hardware and more complex protocols can reduce idle and unintended listening in random access networks.