Adaptive, Direction-Aware Data Dissemination for Diverse Sensor Mobility

A. Boukerche\textsuperscript{1} D. Efstathiou\textsuperscript{2} S. Nikoletseas\textsuperscript{2,3}

\textsuperscript{1}PARADISE Research Laboratory, University of Ottawa, Canada

\textsuperscript{2}Computer Engineering & Informatics Dept. University of Patras, Greece

\textsuperscript{3}Computer Technology Institute (CTI) Patras, Greece

Tuesday, 1 November 2011

In several emerging applications, sensors are embedded in everyday objects
- smart phones, PDAs etc.
- vehicles
- smart clothes

Mobility is a dominant characteristic of the system.
- Sensor nodes are attached to moving objects
- Movement is uncontrollable
- Topology, connectivity changes → Multihop routing is extremely expensive


Main idea:
- mobility replaces connectivity
- “fast” nodes are more capable at ferrying data
- “slow” nodes have to transmit their data in order to accelerate data propagation

Our approach:
- we propose a new (locally computable) network parameter, the direction-aware mobility level
- we exploit sensory mobility as a low energy replacement for connectivity and data propagation redundancy
- we propose a progress-sensitive message flooding inhibition scheme
A number of \( n \) ultra-small homogeneous sensor devices are spread in an area \( D \times D \).

- random uniform initial deployment

Each sensor has a limited cache memory for storing pending messages.

Sensors can roughly estimate their position.

There is one immobile sink located at \( (D/2, D/2) \).
Sensors are attached on moving objects
- Mobility function $\mathcal{M}$.
- Nodes generally follow different mobility functions.
- The movement of each sensor node $i$ at time $t$ is characterized by a mobility level $\mathcal{M}_i(t)$.

Energy Dissipation due to communication of a $k$-bit message at distance $r$

$$E_T = f(k, r^2) \quad E_R = f(k)$$

Data at each device is generated at $\lambda$ messages per second
Estimating Mobility Level

- every $t_l$ seconds node $i$ records its speed and the angle $d_i(t)$ between its direction of movement and the line connecting the current position
- Let $v_i(t)$ be the exponential weighted moving average (EWMA) speed of the last $K$ samples
- Let $d_i(t)$ be the EWMA direction of the last $K$ samples

The mobility level of node $i$ at time $t$ is calculated as:

$$Ml_i(t) = v_i(t) \times \left(1 - \frac{d_i(t)}{\pi}\right)$$
every $t_l$ seconds node $i$ records its speed and the angle $d_i(t)$ between its direction of movement and the line connecting the current position.

Let $v_i(t)$ be the exponential weighted moving average (EWMA) speed of the last $K$ samples.

Let $d_i(t)$ be the EWMA direction of the last $K$ samples.

The mobility level of node $i$ at time $t$ is calculated as:

$$Ml_i(t) = v_i(t) \ast \left(1 - \frac{d_i(t)}{\pi}\right)$$
every \( t_l \) seconds node \( i \) records its speed and the angle \( d_i(t) \) between its direction of movement and the line connecting the current position

Let \( \nu_i(t) \) be the exponential weighted moving average (EWMA) speed of the last \( K \) samples

Let \( d_i(t) \) be the EWMA direction of the last \( K \) samples

The mobility level of node \( i \) at time \( t \) is calculated as:

\[
Ml_i(t) = \nu_i(t) \ast \left(1 - \frac{d_i(t)}{\pi}\right)
\]
Intuition:

- Fast nodes that move in “good” direction are more appropriate for message **ferrying**

- Slow nodes that move in “bad” direction should choose:
  - to transmit data **redundantly** to a number of direct neighbors
  - or to make a long **jump** by transmitting data to a long neighbor

- Redundant transmissions increase the likelihood of message delivery

- With the “expensive” jump transmission we can overcome the “trap” consisting of nodes with low mobility level and make large progress towards the sink
General dissemination protocol:

- **Disconnected operation (no contact to sink):**
  - New events or received messages enter a forward queue.
  - Decision Criterion: Node pops the next message from the front of the forward queue and decides to act suitably according to our decision criterion which is described in following slides.
  - Each message enters the forward phase only once.

- **Connected operation (within range from sink):**
  - Queued messages are forwarded to the sink.

- Queues are FIFO.
Decision Criterion:
The node pops the next message from the forward queue and decides to act suitably according to the following scenarios:

- **Data Ferrying**: If the node has high mobility level, it decides to ferry/carry the data.

- **Data Transmission**: If the node is not ideal to ferry/carry the data, it transmits data using one of the following choices:
  - **Redundancy**: If at least one direct neighbor of the node has a high mobility level, the node disseminates the data to $\beta$ of its neighbors.
  - **Jump**: If all of the node’s direct neighbors have low mobility level, the node transmits to a neighbor TRi-hop closer to sink in order to avoid the “trap” (“bad” neighborhood).
We define $\gamma$ for node $i$ and for its neighborhood, in terms of the current and maximum mobility levels as follows:

$$\gamma_i = \frac{Ml_i(t)}{Ml_{max}(t)}$$

$$\gamma_{neigh i} = \frac{Ml_{max\ neighbor}(t)}{Ml_{max}(t)}$$

1) **Ferrying**: Node $i$ decides to ferry data based on hard threshold $\gamma_i$. If:

$$\gamma_i \geq \frac{1}{2}$$

then ferrying is chosen.
2) **Transmission**: If $\gamma_i < \frac{1}{2}$ then no ferrying is done. Below we propose two methods for deciding whether the node will redundantly propagate data or forward to TR$_i$-hop remote neighbors.

   a) **Hard threshold**: Node $i$ selects to forward data to $\beta_i$ neighbors or to TR$_i$-hop neighbors based on hard thresholds $\gamma_i$ and $\gamma_{neigh} i$.

      1. **Redundancy**: The condition that has to be satisfied for transmitting to $\beta_i$ neighbors is the following:

         $$\gamma_{neigh} i \geq \frac{1}{10}$$

      2. **Jump**: The condition that has to be satisfied for transmitting to TR$_i$-hop neighbors is the following:

         $$\gamma_{neigh} i < \frac{1}{10}$$
b) **Probabilistic**: Let $p_i$ the probability of transmitting to $TR_i$-hop neighbors, and $1 - p_i$ the probability of transmitting to $\beta_i$ direct neighbors; the following choice is perform in node $i$:

\[
Transmission = \begin{cases} 
\text{Redundancy}, & 1 - p_i \\
\text{Jump}, & p_i 
\end{cases}
\]

Where

\[
p_i = \left(1 - \frac{Ml_i(t)}{Ml_{max}}\right) \cdot \left(1 - \frac{Ml_{neigh i}(t)}{Ml_{max}(t)}\right)
\]
Calculation of data redundancy $\beta$

We propose two methods for selecting the number of neighbors $\beta$ to disseminate a message:

a) **Completely local protocol**:

$$\beta_i = \left\lceil \left(1 - \frac{Ml_i(t)}{Ml_{max}}\right) \cdot \left(\frac{D_i}{D}\right) \cdot \delta_1 \right\rceil$$

Where $D_i$ is the distance from sensor $i$ to the sink, and $\delta_1$ represents the maximum redundancy: $\delta_1 = \left\lfloor \frac{dist_{sink}(j)}{R} \right\rfloor$

$$ML_{max} = \nu_{max} \cdot 1 = \nu_{max}$$

- $0 \leq \beta_i \leq \delta_1$
- When $Ml_i$ increases, $\beta_i$ decreases, and vice versa.
- When $D_i$ increases, $\beta_i$ increases, and vice versa.
b) **Neighbor discovery protocol**: 

\[ \beta_i = \left[ \left( 1 - \frac{Ml_{i}^{avg}(t)}{Ml_{max}} \right) \cdot \left( \frac{D_i}{D} \right) \cdot \delta_1 \right] \]

\( Ml_{i}^{avg}(t) \) captures the mobility at the neighborhood of node i at time t.

\[ Ml_{i}^{avg}(t) = \frac{\sum_{j \in \text{neigh}_i(t)} Ml_j(t)}{|\text{neigh}_i(t)|} \]

- \( 0 \leq \beta_i \leq \delta_1 \)
- When \( Ml_{i}^{avg} \) increases, \( \beta_i \) decreases, and vice versa.
- When \( D_i \) increases, \( \beta_i \) increases, and vice versa.

The rationale is to calculate large values of \( \beta \) for “slow” moving in “bad” direction and distant from the sink nodes. The opposite happens for “fast”, moving in “good” direction and close to the sink nodes.
Calculation of length of jump $TR_i$

$$TR_i = \left\lceil \left( 1 - \frac{Ml_i(t) + Ml_i^{avg}(t)}{Ml_{max}} \right) \cdot \left( \frac{D_i}{D} \right) \cdot \frac{\delta_1}{2} \right\rceil$$

- $0 \leq TR_i \leq \frac{\delta_1}{2}$
- When $Ml_i^{avg}$ increases, $\beta_i$ decreases, and vice versa.
- When $D_i$ increases, $\beta_i$ increases, and vice versa.

The rationale is to calculate large values of $TR_i$ for “slow”, moving in “bad” direction, distant from the sink nodes which are in relatively “bad” neighborhood.
Neighbor Selection

Our protocol has to do neighbor selection in two cases, when selecting:

- direct neighbors in order to do redundancy
- when jumping to a long neighbor so as to avoid bad neighborhood
- **Completely Random Selection.** Select $\beta_i$ random neighbors.

- **Fittest Candidate Selection.** Select $\beta_i$ neighbors such that $Ml_i(t) < Ml_j(t)$

- **Probabilistic Candidate Selection.** Select $\beta_i$ neighbors with probability $p_j$

$$p_j = \begin{cases} \frac{Ml_j(t)}{Ml_i(t)} & Ml_j(t) \leq Ml_i(t) \\ 1 & Ml_j(t) > Ml_i(t) \end{cases}$$
• **Completely Random Selection.** Select $\beta_i$ random neighbors.

• **Fittest Candidate Selection.** Select $\beta_i$ neighbors such that $Ml_i(t) < Ml_j(t)$
Inhibition of obsolete messages

Each message carries a hop counter $h_c$. The decision to whether to propagate or just store the message is done as below:

$$Decision = \begin{cases} 
    \text{Propagate message} & h_c < h_{opt} \\
    \text{Store message} & h_c \geq h_{opt}
\end{cases}$$

where $h_{opt} = \left\lceil \frac{\text{dist}_{sink}(j)}{R} \right\rceil$.

If $\text{dist}_{sink}(j)$ is unknown, let $\text{dist}_{sink}(j) = \frac{D}{2}$.

The inhibition decision depends on the distance the message has traveled with respect to the overall required distance.
By choosing appropriate mobility models and parameters we define 4 mobility patterns each resembling a particular mobility role: \( M_{work} \), \( M_{walk} \), \( M_{bic} \), \( M_{veh} \).
Mobility Transitions

- Mobility role may vary with time
- Mobility transition diagram
  - Each vertex represents a mobility model
  - Edges are associated with a probability of transition
Network Simulator (ns-2 version 2.33) and TRAILS extensions

- We implement a flooding protocol $\beta_i = \infty$
- We implement a gossiping protocol
- We implement our adaptive Direction Sensitive Mobility Protocol (DSMP)
- We implement Adaptive Mobility Protocol (AMP) which is an other adaptive redundancy protocol
- $1000m \times 1000m$
- $\lambda = 0.025$ events/sec, 20 events per sensor
- default transmission range $R = 70m$
- Sink is positioned at $(500, 500)$
Node densities:
- 300 for the first set of experiments
- 50, 100, 150, 200, 250 and 300 for the second set of experiments

Node movement:
- 25% follow $M_{work}$, 25% follow $M_{walk}$, 25% follow $M_{bic}$, 25% follow $M_{veh}$
- 25% follow $C_1$, 25% follow $C_2$, 25% follow $C_3$, 25% follow $C_4$

We measure
- Success Rate $P_{rs}$
- Energy Dissipated $E_{tot}$
- Delivery Delay $D$
Experimental Setup 3/3

Two set of experiments:

1st set We compare our DSMP protocol with flooding, gossiping and AMP

2nd set We investigate the impact of density to the two adaptive protocols (AMP, DSMP)
Gossiping achieves the lowest success rate because of its unbound randomness.

Flooding achieves low success rate due to the limited buffers.

The adaptive protocols (AMP and DSMP) achieve highest success rate.
Flooding consumes the highest amount of energy.

Our DSMP protocol consumes about 22% more energy than AMP protocol, due to long transmissions.
Our DSMP protocol achieves lower latency.

Flooding achieves low latency, but higher than our DSMP protocol.

Gossiping achieves the highest average delay among all protocols.
Our DSMP protocol behaves better than AMP protocol in sparse networks as well.

Over 100 nodes, DSMP has a very satisfactory success rate over 90%.
For small node densities (lower than 150 nodes), DSMP protocol consumes less energy than AMP.
For any node density, DSMP’s delay is by far lower than AMP’s. This is due to its ability to “jump” over sparse or even disconnected areas.
Conclusions & Future Work

- the mobility parameter captures the ability of a node to arrive close to the sink quite fast
- ferrying serves as a low cost replacement for data dissemination
- in the case of "low" mobility either:
  1. data propagation redundancy is increased
  2. long-distance data transmissions are used to accelerate data dissemination

Extensions

- multiple (static) sinks
- mobile sink(s)
- more regular, human-related mobility patterns
- comparison with other ferrying and opportunistic routing mechanisms