Course “Algorithmic Foundations of Sensor Networks”
Lecture 8: Node Mobility in Sensor Networks

Sotiris Nikoletseas
A collection of small, autonomous, battery-operated sensor devices.

Each sensor device integrates sensing, computing, and communication capabilities.

Sensor devices are deployed in a non-sophisticated way (dropped, thrown, positioned).

Sensor devices collaborate to perform a set of sensing tasks.

Measured data are reported to a Control Center (sink).
• much larger number of devices
• more dense deployment / interactions / complexity
• more limited capabilities
• prone to failures
• more frequently/more dynamically changing topology
• less global knowledge
Motivation

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

Environmental mobility (water flow, wind etc.)

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 and may be even infeasible
Vehicular Ad-Hoc Networks (VANETs)
Smart Shirt Sensory Architecture

- Microphone
- Optical Fiber
- Basic Grid
- Data Bus
- Sensor
- Multi-function Processor
**Sensatex™ Smart Shirt**

The Smart Shirt Wearable Motherboard™ is a flexible, wearable open platform that can be customized to monitor vital signs, external impact, and other data through sensors woven into its fabric. Here's how the system works:

1. The Sensatex™ Smart Shirt is worn during any activity. Embedded sensors monitor heart rate, respiration, and other vital signs in a customizable fashion.

2. Data is sent via satellite or cellular tower from the Smart Shirt™ processor to an information hub.

3. The information hub constantly monitors the Smart Shirt™ wearer's vital signs for specific job or health-related hazards.

4. If a problem arises, an emergency medical hub is immediately alerted. Paramedics can reach the Smart Shirt™ wearer quickly, already informed about his condition and able to attend to his needs.

5. Data continually travels to a secure internet site where the wearer can log on anytime, anywhere to review.

**Other Applications**

As an open platform, the Smart Shirt Wearable Motherboard™ is customizable to fill different monitoring needs.

- Infants can be monitored for sleep apnea and other infant disorders.
- Firemen can be monitored for smoke-inhalation and alerted when in danger.
- Geriatric and post-operative monitoring offers a greater sense of security and improves quality of life.
- Monitoring in police and military applications can enhance job safety and performance.

Electrical and optical conductive fibers are woven or knitted with common textile fibers and connected to the data bus via the Sensatex™ Interconnection Technology.
Mobile sinks/nodes exchange messages only when communicating directly or over few hops.

New Challenges

- Longer delivery delays
- Bad scalability when network area increases
- Routing and localization problems become more difficult
Advantages of Sink Mobility

- Sparse, disconnected and irregular networks can be better handled
- Communication obstacles can be bypassed
- Better load distribution
- Scales well with respect to number of sensors
- Reduces communication distance
- Reduces energy consumption on the sensors → System lifetime increases
Moving elements:
- Sink(s)
- Sensor nodes
- Relay nodes
Controlled Vs. Uncontrolled Mobility

Controlled Mobility

+ Topology Adaptivity
+ Using controlled mobility, resources can efficiently be moved to regions where they are required (Increase lifetime).
+ Delay can be bounded using controlled mobility
  - Cost of mechanical movement (robots)
  - Complex protocols

Uncontrolled Mobility

+ More “simple” protocols
  - No guarantees for data delivery latency
  - Non-coverage of certain network areas
Paper

What is ZebraNet? Why ZebraNet?

- Collaboration research work between wildlife biologists and mobile network computer scientists
- Tracking nodes (collars) with GPS, Flash Memory, wireless (radio) transceiver, small CPU
- There is no cellular infrastructure in the area of interest
- Peer-to-peer data communication
- Wireless sensor network for wildlife tracking
Design Considerations

- mobile base station
- moving nodes with unknown mobility models
- energy trade-offs
It is important to learn:

- How human development into wilderness areas affects indigenous species there
- What are the migration patterns of wild animals and how they may be affected by changes in weather patterns or plant life

In order to learn such details about animals requires both detailed long-term position logs as well as other biometric data such as heart rate, body temperature, and frequency of feeding.
Many previous studies rely on collaring a sample subset of animals with simple VHF transmitters.

Researchers periodically drive through an area with a receiver antenna and listen for pings from previously-collared animals.

Limitations of the above approach:

- Data collection is infrequent and may miss many interesting events
- Data collection is often limited to daylight hours
- Data collection is impossible or severely limited for reclusive species that avoid human contact
More sophisticated trackers use global positioning systems (GPS) to track position and use satellite uploads to transfer data to a base station.

Limitations of the above approach:

- Satellite uploads are slow and power-hungry
- Downloads of data from the satellite to the researchers are both slow and expensive
Design Goals

Zoologists’ requirements

- GPS position samples, every 3 minutes
- Activity logs, taken 3 minutes every hour
- 1 year of operation with no human intervention
- Operate over thousands of square kilometers
- No fixed base station, antennas, cellular network
- High delivery rate of data logs (Latency is not critical)
- Limited collar weight (e.g. 3 -5 lbs for zebra collar)

Implications to design

- Weight limit, energy limitation
- Transmission range
- Storage capacity
Effect of Mobility

- Nodes (collars) fitted on zebras
- To understand node mobility requires understanding of how fast, in what direction and with what forces of attraction/repulsion zebras move.
- Movement patterns: grazing, graze-walking, fast-moving
• Every sample indicates the net distance traveled in a 3 minute interval
• Zebras tend to move very slowly, as they spend most of their time simply grazing
Design goals

- Total weight 3-5 lbs
- Energy 5 days of no recharge
- Battery rechargeable using solar cell

Amount of data

- 30 coordinates per hour
- 240 bytes per hour
- 1 Collar-day 6KB
GPS, Short/Long Radio, Flash Ram and CPU.
ZebraNet Characteristic

- Not every collar is within range of the base station (hop by hop communication)
- The nodes (collar) move around almost constantly
- Base station is also mobile
- Base station is active from time to time
- High success rate is important (latency is not critical)

Protocol Strategies

- Flooding protocol
- History-based protocol
Flooding protocol

- Flood data to all neighbors whenever they are discovered
- Nodes that move extensively and meet a fair number of the nodes (highly - interactive), then given enough time, data will eventually migrate back to the base
- Base station does not necessarily have to come into contact with all the nodes in the system (by identifying a few highly - interactive nodes, we can collect a substantial amount of data readily)
- Large amount of data can lead to excessive demands for bandwidth, storage and energy
History-based protocol

- After peer discovery, choose at most one peer to send to per discovery period: the one with best past history of delivering data to base.
- Can reduce amount of data in network.
- ZebraNet is very dynamic (both collars and base station are mobile). Then, this protocol may mis-direct traffic and get a poor success rate.
Experimental Results

The most important factors
- storage
- bandwidth

The most important metrics
- success rate
- energy
Two metrics of connectivity

- Direct connectivity
- Indirect connectivity
Storage Constraints

We assume constrained storage and infinite bandwidth.

- The peer-to-peer protocols perform better than the direct protocol. This is surprising since these protocols require that the storage handle both the collar’s own data and of their peers.
- The deletion strategy followed prioritizes a node’s own data over others. At worst, a protocol stores only its own data.
Energy Tradeoffs

- While flooding makes sense at low-radio-range and low-connectivity points, it is a poor choice for the high-connectivity regime.
- The history-based and the direct protocol have similar performance.
Delay tolerant networks (DTNs) are occasionally-connected networks that may suffer from:

- Network Partitioning
- Network Interruptions, Failures and Heterogeneity
- Asymmetric, Long and Variable Data-Rates
- Energy, Bandwidth, Buffer and Cost Restrictions

Representative DTNs include:

- Remote Area Networks
- Military Battlefield Networks
- Mobile Sensor Networks
Keyword: Epidemic Routing (flooding in a disconnected context)

Goal is to deliver messages with high probability even when there is never a fully connected path.
The overall goal of Epidemic Routing is to

- maximize message delivery rate
- minimize message latency
- minimize the total resources consumed in message delivery
Main idea:

- mobility replaces connectivity
- message ferrying replaces multihop data propagation
- “fast” nodes are more capable at delivering data
Our approach:

- we propose a novel network parameter, the mobility level
- we exploit sensory mobility as a replacement for connectivity and data propagation redundancy
- we propose adaptive protocols, that propagate less data in the presence of high, diverse mobility and favor relay-sensors with higher mobility levels
- we propose a mobility-and progress-sensitive probabilistic message flooding inhibition scheme that further reduces communication cost
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 $\mathcal{ML}_i(t)$.

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
every $\delta$ seconds node $i$ records its position $(p_0, p_1, \cdots, p_k)$ and speed $(\upsilon_0, \upsilon_1, \cdots, \upsilon_k)$

After recording $K$ measurements, a series of distances from $p_0$ is calculated: $(d_0, d_1, \cdots, d_k)$

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

$$Ml_i(t) = \upsilon_i(t) \cdot d_{i_{max}}(t)$$
every $\delta$ seconds node $i$ records its position $(p_0, p_1, \cdots, p_k)$ and speed $(v_0, v_1, \cdots, v_k)$

After recording $K$ measurements, a series of distances from $p_0$ is calculated: $(d_0, d_1, \cdots, d_k)$

The mobility level of node $i$ is:

$$Ml_i(t) = v_i(t) \cdot d_i^{max}(t)$$
Dissemination Protocols

Intuition:
- By disseminating messages between nodes the likelihood of delivery increases.
- “Faster” nodes cover the network faster and have increased probability of delivering messages.

General dissemination protocol:
- Disconnected operation (no contact to sink):
  - New events or received messages enter a forward queue.
  - Dissemination: each sensor forwards a message from the head of the forward queue to $\beta$ neighbors.
  - Ferrying: forwarded messages are stored in a delivery queue.
  - Each message enters the forward phase only once.
- Connected operation (within range from sink):
  - Queued messages are forwarded to the sink.
- Queues are FIFO.
Assume each node is bounded by $v_{\text{max}}$, then:

$$Ml_{\text{max}} = v_{\text{max}} \cdot (v_{\text{max}} \cdot t_l)$$

where $(v_{\text{max}} \cdot t_l)$ gives the maximum dislocation achieved during time $t_l$.

**Completely local protocol:**

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

- When $Ml_i(t)$ increases, $\beta_i$ decreases, and vice versa.
- $0 \leq \beta_i \leq \frac{D}{Ml_{\text{max}}}$
Neighbor discovery: Each node transmits a query when needed, other nodes reply by sending their $Ml_i(t)$. Let $\text{neigh}_i(t)$ the set of all neighbors of node $i$ at time $t$

\[
Ml_i^{avg}(t) = \frac{\left( \sum_{j \in \text{neigh}_i(t)} Ml_j(t) \right) + Ml_j(t)}{|\text{neigh}_i(t)| + 1}
\]

**Limited awareness protocol :**

\[
\beta_i = \left\lceil \frac{D}{Ml_{max}} \cdot \left( 1 - \frac{Ml_i^{avg}(t)}{Ml_{max}} \right) \right\rceil
\]

- When $Ml_i^{avg}$ increases, $\beta_i$ decreases, and vice versa.
- $0 \leq \beta_i \leq \frac{D}{Ml_{max}}$
- **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 $p_j$

\[
p_j = \begin{cases} 
\frac{Ml_j(t)}{Ml_i(t)} & Ml_j(t) \leq Ml_i(t) \\
1 & Ml_j(t) > Ml_i(t) 
\end{cases}
\]
Our 2nd approach

Paper
Our approach

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
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 :

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

$$Ml_i(t) = v_i(t) \times \left(1 - \frac{d_i(t)}{\pi}\right)$$
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:

\[
Ml_i(t) = v_i(t) \times \left(1 - \frac{d_i(t)}{\pi}\right)
\]
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).
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: $\delta_1 = \left\lfloor \frac{\text{dist}_{sink}(j)}{R} \right\rfloor$

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

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

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

\(Ml_{i}^{avg}(t)\) 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 $TR_i$

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

- $0 \leq TR_i \leq \frac{\delta_1}{2}$
- 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.
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 $p_j$

$$ p_j = \begin{cases} 
\frac{Ml_j(t)}{Ml_i(t)} & Ml_j(t) \leq Ml_i(t) \\
1 & Ml_j(t) > Ml_i(t) 
\end{cases} $$
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 $D$
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.
Conclusions

- 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
“Aggregated Sensory Data Collection by Mobility-based Topology Ranks”, Angelopoulos Constantinos Marios, Nikoletseas Sotiris, GLOBECOM 2009 - PE-WASUN 2009 (new version with aggregation)
In a WSN with *full mobility scheme*, where both sensors and sink move dynamically, how can the sink *efficiently* collect data from sensors?
The Problem
Modelling Dynamic Sensory Mobility

C1: Transitions between slow mobility roles

C2: Transitions for medium mobility level

C3: Transitions for medium mobility with fast bursts

C4: Transitions for fast mobility
Our Approach

- Sensors are moving inside the network area, independently of each other.
- Each sensor periodically measures the number of its neighbors and stores a triplet.
- Each triplet consists of: the number of neighbors, a timestamp, the current position.

Based on triplets, the sink is going to exploit general topological information.
Each triplet is assigned a value via the ranking function

\[ R = \frac{d_{\text{local}}^2}{\Delta P \Delta T} \]

where:

- \( d \) is the number of neighbors.
- \( \Delta P \) the distance between position where \( d \) was measured and current position.
- \( \Delta T \) the time interval when \( d \) was measured and current time.
Our Approach II

Each triplet is assigned a value via the ranking function

\[ R = \frac{d_{\text{local}}^2}{\Delta P \Delta T} \]

- Only one triplet is stored each time.
- A new triplet replaces stored one in sensor memory, **iff** it is assigned a higher value.
- When a sensor reaches the radio range of the sink, along with the sensory data, the stored triplet is also sent.
- The sink chooses to move towards the direction corresponding to the highest ranking collected triplet.
The sink initially traverses the network area at random direction. For a short period of time it collects triplets. Triplets A,B corresponding to positions relatively close are aggregated into C, based on an $\text{angle}_{\text{thresh}}$

\[
\begin{align*}
C_{d_{\text{local}}} & = A_{d_{\text{local}}} + B_{d_{\text{local}}} \\
C_P & = \frac{A_P * A_{d_{\text{local}}} + B_P * B_{d_{\text{local}}}}{A_{d_{\text{local}}} + B_{d_{\text{local}}}} \\
C_T & = \frac{A_T * A_{d_{\text{local}}} + B_T * B_{d_{\text{local}}}}{A_{d_{\text{local}}} + B_{d_{\text{local}}}}
\end{align*}
\]
Performance Findings

- Latency improves up to 8 times
- Better success rate, from 93% to 98%
- Slightly (10%) more energy dissemination
- Improvements also in homogeneous placement


