Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN)

Yu Wang and Hongyi Wu
Center for Advanced Computer Studies
University of Louisiana at Lafayette
Typical Applications and Unique Characteristics of DFT-MSN

- Applications:
  - Flu virus tracking
  - Air quality monitoring
  - Wild animal monitoring

- Unique characteristics:
  - Nodal mobility
  - Sparse connectivity
  - Delay/fault tolerability
  - Limited buffer

- Mainstream approaches of sensor networks may not work effectively.
DFT-MSN: Architecture

An overview of the integrated self-configurable wireless mesh network and delay/fault-tolerant mobile sensor system. $S_1$-$S_{10}$: sensors; HES$_1$-HES$_2$: high end sensors (sinks); AP$_1$-AP$_9$: access points of backbone network. Only $S_2$ and $S_3$, $S_4$ and $S_5$, and $S_6$ and HES$_2$ can communicate with each other at this moment.
Related Work

- Delay Tolerant Network (DTN)

- DTN in Sensor Networks
  - Static sensor nodes and sinks: e.g., Ad hoc Seismic Array, etc.
  - Static sensor nodes and mobile sinks: Data Mule.
  - Mobile sensor nodes and/or sinks: ZebraNet, SWIM, DFT-MSN.
Studies of Two Basic Approaches

- Direct Transmission
  - M/G/1 Queuing Model
  - Assume message generation is Poisson
  - Lemma 1: the service time of the message is Pascal distributed

- Simple Flooding
  - Analyze flooding overhead, delay, and delivery probability

- Optimized Flooding
  - Estimate message delivery probability and terminate flooding
  - To reduce flooding overhead and energy consumption
Studies of Two Basic Approaches

- Analytical models are verified via simulations
An Overview of The Proposed DFT-MSN Data Delivery Scheme

The proposed Fault Tolerance-based Adaptive Delivery Scheme (FAD) is based on two key parameters:

- The nodal delivery probability:
  - Assisting data transmission.
  - The metrics for when and where to transmit data message

- The message fault tolerance
  - Assisting queue management.
  - The metrics for which messages to transmit or drop
The delivery probability indicates the likelihood that a sensor can deliver data messages to the sink. The delivery probability of a sensor $i$, $\xi_i$, is updated as follows,

$$
\xi_i = \begin{cases} 
(1 - \alpha)[\xi_i] + \alpha \xi_k, & \text{Transmission} \\
(1 - \alpha)[\xi_i], & \text{Timeout}, 
\end{cases}
$$

where $[\xi_i]$ is the delivery probability of sensor $i$ before it is updated, $\xi_k$ is the delivery probability of node $k$ (a neighbor of node $i$), and $0 \leq \alpha \leq 1$ is a constant employed to keep partial memory of historic status.
Message Fault Tolerance

- The fault tolerance of a message is defined to be the probability that at least one copy of the message is delivered to the sink by other sensors in the network.

- Considering a sensor $i$ multicasting a data message $j$ to $Z$ nearby sensors, the message transmitted to sensor $\psi_z$ is associated with a fault tolerance of $\mathcal{F}_\psi^j$,

$$\mathcal{F}_\psi^j = 1 - (1 - [\mathcal{F}_i^j])(1 - \xi_i) \prod_{m=1, m \neq z}^{Z} (1 - \xi_{\psi_m}),$$

where $[\mathcal{F}_i^j]$ is the fault tolerance of message $j$ at sensor $i$ before multicasting.

The fault tolerance of the copy at sensor $i$ is also updated accordingly using similar calculation.
Data Transmission

- Data transmission decision is made based on the nodal delivery probability.
- First step: learns the neighbors’ delivery probabilities and available buffer spaces via simple handshaking messages.
- Second step: sends the message to a set of neighbors with higher delivery probabilities, and at the same time, controls the total delivery probability of that message just enough to reach a predefined threshold, in order to reduce unnecessary transmission overhead.
Queue Management

- The queue management scheme is based on the fault tolerance.
- Message with the smallest fault tolerance is always at the top of the queue and transmitted first.
- Message dropping happens in two situations:
  - The queue is full.
  - The fault tolerance of a message is larger than a threshold.
## Simulations

### Simulation setup

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum sensor transmission range</td>
<td>$10 \text{ m}$</td>
</tr>
<tr>
<td>Number of sensor nodes</td>
<td>100</td>
</tr>
<tr>
<td>Number of sink nodes</td>
<td>3</td>
</tr>
<tr>
<td>Size of network area</td>
<td>$200 \times 200 \text{ m}^2$</td>
</tr>
<tr>
<td>Size of a zone</td>
<td>$40 \times 40 \text{ m}^2$</td>
</tr>
<tr>
<td>Probability to move out of a zone</td>
<td>20%</td>
</tr>
<tr>
<td>Probability to move back to home zone</td>
<td>100%</td>
</tr>
<tr>
<td>Maximum queue length</td>
<td>200</td>
</tr>
<tr>
<td>Message generation rate</td>
<td>0.01/s</td>
</tr>
<tr>
<td>Message length</td>
<td>50 bits</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>2500 bps</td>
</tr>
<tr>
<td>Nodal moving speed</td>
<td>0 – 5 m/s</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Simulations

- Update of delivery probabilities

(a) Initial deployment.

(b) 1000 seconds later.
Simulations

- Impact of number of sink nodes

(a) Average delivery ratio.

(b) Average delay.
Simulations

- Impact of maximum queue length

(a) Average delivery ratio.

(b) Average delay.
Simulations

- Impact of nodal speed

(a) Average delivery ratio.

(b) Average delay.
Follow-up Work

- A generic queuing model for delay tolerant mobile networks
- Prototype and experimental testbed (Percom’06 PerSeNS workshop)
- An alternative approach based on Erasure coding (Percom’06 Ubicare workshop)
Conclusion

- DFT-MSN is proposed for pervasive information gathering
- DFT-MSN has several unique characteristics, such as nodal mobility, sparse connectivity, delay/fault tolerability, and limited buffer
- Studied two basic approaches based on queuing theories
- Proposed an efficient message delivery scheme
- Simulated the proposed data delivery scheme, showing high delivery ratio and low transmission overhead
- In our follow-up work, we have proposed an alternative approach, carried out deep analytic studies, developed a small-scale testbed