Network Models for Molecular Communication

Ayush Dubey, Shretima Tandon
Supervisor: Prof Aaditeshwar Seth
Nanonetworks

- Expand capabilities of single nanomachines by allowing them to cooperate and share information.

- Molecular communication stands out as the most promising approach due to its advantages in terms of size, biocompatibility and biostability and energy efficiency.

Applications
- targeted drug delivery vehicles, e.g. nanomachines deployed for destruction of cancer cells
- biodegradation of unwanted materials
- removal of bacteria etc from food items and water
- bio-detection devices
Our goal is to provide a high-level interface for programming molecular communication applications on nanomachines.
Related work

- Three modes of molecular communication have been proposed
  - emulating intra-cellular communication along molecular motors
  - inter-cellular communication via calcium signaling
  - long distance communication using pheromones

- Five basic components - encoding, sending, propagating, receiving and decoding

- Modeling and simulating information rate, channel noise

- Mike Moore’s simulator for diffusion and molecular motors
# Basic Nanomachine Functionalities

<table>
<thead>
<tr>
<th>Ca(^{2+}) signaling</th>
<th>Molecular motors</th>
<th>Generic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Callbacks</strong></td>
<td><strong>Actions</strong></td>
<td><strong>Callbacks</strong></td>
</tr>
<tr>
<td>Sense Ca(^{2+}) frequency / conc., changes that result from this, identify receptors / ligands on cell surface</td>
<td>Decode information in received molecule, decode molecular properties (polarity, orientation), release payload on motor, amplify received molecule, construct microtubule</td>
<td>Sense specific cells / tissues, environment changes, receive count from clock, external radio signals, sense dye tagged nanoparticles</td>
</tr>
<tr>
<td>Stimulate neighborhood by IP(_3), release Ca(^{2+}) at particular conc. / frequency, establish / change permeability of gap junctions for Ca(^{2+})</td>
<td>Encode information in DNA or molecular properties (polarity, orientation), release payload on motor, amplify received molecule, construct microtubule</td>
<td>Produce motion, extract / deposit molecule from body, generate onboard power, degrade stored molecules, secrete chemicals</td>
</tr>
</tbody>
</table>

**Callbacks**
- Sense specific cells / tissues
- Environment changes
- Receive count from clock
- External radio signals
- Sense dye tagged nanoparticles

**Actions**
- Produce motion
- Extract / deposit molecule from body
- Generate onboard power
- Degrade stored molecules
- Secrete chemicals
Network structure

- Rarely need to program at individual nanomachine level
- Agglomerate many nanobots into a *quorum* – the nodes of our network – increase efficacy, distribute workload
- Each quorum consists of different types of sensors, actuators, and special nanomachines (next slide)
- One unique identification number for each quorum
- Flat hierarchical addressing scheme
  - `<quorum no> . <machine type>`
  - `<quorum no> . *`
  - `* . <machine type>`
- Primitives are callable at quorum level only
  - e.g. QuorumA.SendMessage
  - However code running on individual nanomachines
Quorum layout

- Spherical configuration
  - Sensors (yellow) at boundary, actuators (blue, green) in the centre
  - Intra-quorum communication: $Ca^{2+}$ signaling (always broadcast)
  - Actuators placed either randomly or in layers in order of release (determined by initially parsing code)
- Specialized nanomachines (red) on the periphery for
  - Routing
  - Linking
Quorum layout (2)

- Cuboidal configuration
  - Easier to decipher direction in case of MoveQuorum message
  - Intra-quorum communication: molecular motors
  - Each nanomachine inside the cuboid connected to 6 others
  - Routing/linking nanomachines on the edges
  - Sensors over the faces
Middleware primitives

- **ConfigNetwork**
  - Setup initial environment and store in nanomachine
  - Determine pheromone usage, mode of communication, area by parsing code

- **SetupQuorum**
  - Input: list of nanomachines, quorum ID
  - Setup quorum as per layout decided by parsing code

- **MoveQuorum**
  - If nano-compass available onboard, can move to specific coordinates
  - Or can follow a pheromone trail left by another quorum

- **MakeLink**
  - Makes an edge in the network between 2 quorums
  - Makes use of linking nanomachines
Middleware primitives (2)

**SendMessage**
- Notify routing nanomachines, which then forward the message on their outgoing link
- Can introduce reliability by retransmissions, acks – depending on simulation results

**Sense Environment**
- Method for sensing environmental parameters like density, special types of molecules
- Calls appropriate function

**Release Payload**
- Actuator nanomachines release payload
- Subsequently detach from quorum
Middleware primitives (3): an example

Algorithm 4.5: $\text{SendMessage}(\text{Msg}, \text{Dest}, \text{QoSparams})$

if This nanomachine is not routing nanomachine
    then $\text{IntraQuorumBroadcast}(\text{SendMessage}(\text{Msg}, \text{Dest}, \ldots))$
else
    (comment: This nanomachine is routing nanomachine)

if Dest present in routing table or Dest is all quorums
    then
        \begin{align*}
        &\{ \\
        &\text{Send the } \text{Msg} \text{ on outgoing link} \\
        &\text{IntraQuorumBroadcast(} \text{Message } \text{“Msg” sent})\\
        &\text{wait for } \text{TIMEOUT} \\
        &\text{if still not received intra quorum confirmation} \\\n        &\text{then} \\
        &\{ \text{comment: Dest not present in any routing table of this quorum} \} \\
        &\text{Send the } \text{Msg} \text{ on outgoing link} \\
        \end{align*}
else
    (comment: Dest not present in any routing table of this quorum)
Other components

- Nanomachine-level functions
  - Used in the middleware code, not available to the programmer as a primitive
  - Called at level of individual nanomachines
  - ReceiveMessage, IntraQuorumBroadcast

- Hardwired Applications
  - Preloaded into every nanomachine
  - Make use of the provided middleware
  - SetupNetwork, GetQuorumState
Example application

- Targeted drug delivery: sensing and destroying diseased cells in the body

function: Release Drug
  ReleasePayload (drug)
  ReleasePayload (pheromone)
  DestroyQuorum

main: Targeted Drug Delivery
  SenseEnvironment (pheromone trail, 5nm, timeout, MoveQuorum(follow pheromone trail))
  SenseEnvironment (diseased cells, 5nm, timeout, Release Drug)
Future work: Simulation

- Objectives
  - Different methods of setting up the graph: user input, preferential attachment, grid-like network
  - Reliability and latency of message transfer in various environments, different modes of communication
  - Different ways to carry out application:
    - have both sensors and actuators in the same quorum
    - have scout quorums with only sensors and pheromones, and action quorums with the drug to be delivered.
  - Different ways of intra-quorum broadcast: $Ca^{2+}$ signaling, sparse rail network, cube shaped quorum

- Models to be simulated
  - Noise: over rail, diffusion of pheromones, movement of $Ca^{2+}$ ions
  - Movement of quorums
  - Release of payload
References


