# Nano-wireless Communication for Microrobotics: bridging the gap

Ludovico Ferranti
University of Roma Sapienza
Rome, Italy
ferranti.1348932@studenti.uniroma1.it

Francesca Cuomo
University of Roma Sapienza, DIET
Rome, Italy
francesca.cuomo@uniroma1.it

## ABSTRACT

Among all the new robotics technologies, great interest is rising in the field of microrobotics. Micro Electro-Mechanical Systems (MEMS) microrobots are miniaturized electro-mechanical devices that can perform various tasks in a wide area of applications. Despite the low-power and low-memory capacity, they are provided with sensors and actuators. Self-reconfiguration is a key factor for MEMS microrobots to perform their tasks and to optimize their communications in order to achieve efficiency, parallelism and scalability. Nano-transceivers and nano-antennas operating in the Terahertz Band offer a promising communication paradigm, providing nanowireless networking directly integrated in MEMS microrobots. Catoms from Claytronics project are an appropriate microrobotics case to explore this novel framework. Several logical topology shape-shifting algorithms have been implemented and tested, along with different nano-wireless simulations. This paper aims to provide a survey on nano-wireless communication for modular robotics and propose some optimization choices. Special emphasis is given to the use of the nano-wireless communications for topology formation and maintenance in microrobotics.

## Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]:
; I.2.9 [**Robotics**]:

## Keywords

MEMS Microrobots; Nano-networks; Terahertz band; Distributed algorithm; Topology Formation

## 1. INTRODUCTION

Micro electro mechanical system (MEMS) is the new frontier of technology that empowers the batch fabrication of miniature mechanical devices. MEMS are miniaturized and low-power devices with the ability to sense and act in their environment. High expectation is put in the mass production of MEMS nodes, making them a very affordable technology. Their small size and the batch fabrication process will allow MEMS microrobots to be used in many areas of everyday life. Usually, MEMS nodes size varies from well below one micron on the lower end of the dimensional spectrum to several millimeters. Their applications require a massive deployment of nodes, thousands or even millions, which define a new concept of Distributed Intelligent MEMS (DiMEMS). A DiMEMS system is composed of a compact swarm of MEMS nodes. Several applications need the ensemble shape to be designed in particular topologies therefore efficient communication among nodes is crucial. The very limited power supply of microrobots makes it difficult to implement a precise movement to reach a certain destination position and it is still one of the major challenges in this field. Although the power consumption for moving has been lowered, communication and computation requirements still represent a challenge in terms of energy. Optimizing the number of movements and message exchanges of microrobots is therefore crucial in order to save energy.

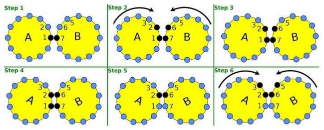


**Figure 1: Catoms moving principle [5]**

Controlling the self-reconfiguration process is very complex because it involves the distributed coordination of large numbers of identical modules connected in time-varying ways. The range of exchanged information and the amount of displacements determine the communication and energy complexity of the distributed algorithm. As soon as the information exchange involves close neighbors, the complexity is low and the resulting distributed self-reconfiguration smoothly scales along with network size. Considering a moderate complexity in message, execution time, number of movements and memory usage finding an optimal configuration is still an open issue and the logical topology of the network has to optimize through rearrangement of the physical topology. Several microrobotics projects are following these assumptions. Harvard University Kilobots project aims at achieving a scalable and self-assembly design for microrobotics [21]. Swarms of Kilobots, composed of thousands of tiny modular microrobots, move through vibration motors on a lucid plane, which is used to reflect infrared beams as a mean of communication. Another advanced instance of microrobotics is the Claytronics project by Carnegie Mellon University and Intel. In all these modular robots the possibility to have wireless communications may facilitate topology formation and maintenance as well as lower the energy

*NANOCOM15* September 21 - 22, 2015, Boston, MA, USA

http://dx.doi.org/10.1145/2800795.2800803.

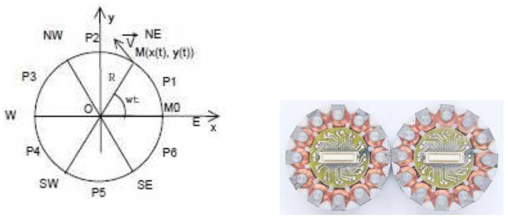
**Table 1: Catoms parameters and characteristics [1]**

| | Macro | Micro | Nano |
|---|---|---|---|
| Dimensions | $> 1\ cm$ | $> 1\ mm$ | $< 10\ \mu m$ |
| Weight | tens of $g$ | hundreds of $g$ | $< 1\ mg$ |
| Power | <2 W | tens of $mW$ | tens of $nW$ |
| Locomotive Mechanism | Programmable Magnets or ElectroMagnets | Electrostatics | Aerosol |
| Adhesion Mechanism | Nanofiber Adhesives or Magnets | Programmable nanofiber adhesives | Molecular surface adhesion and covalent bonds |
| Manufacturing Methods | Conventional manufacturing and assembly | Micro/Nano-fabrication and micro-assembly | Chemically directed self-assembly and fabrication |
| Resolution | Low | High | High |
| Cost | $$$/catom | $/catom | Millicents/catom |

consumption of the microrobots. In the following we focus on Claytronics project as an example and we investigate the use of nano-wireless communications framework. The remainder of this paper is organized as follows: in Section II a brief presentation of Claytronics project is given, underlining how microrobots work and cooperate and the open challenges. Section III deals with nano-wireless communications operating in Terahertz band, Section IV will present an addressing scheme that is feasible with microrobotics constraints and finally Section V will summarize the ideas about using nanowireless communication for microrobotics.

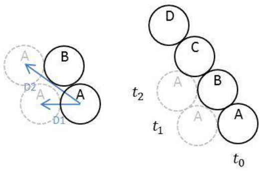
## 2. CLAYTRONICS MICROROBOTS

Claytronics is a concept that combines nanoscale robotics and computer science to create individual nanometer-scale computers called *catoms*, which can interact with each other to form shapes. This idea is more broadly referred to as Programmable Matter. Other approaches to modular robotics used to create a conglomerate of tens or even hundreds of small autonomous robots which move in coordination to achieve a global effect not possible by any single unit [1].



(a) Node modeling, in each movement the node travels the same distance

(b) Two planar catoms

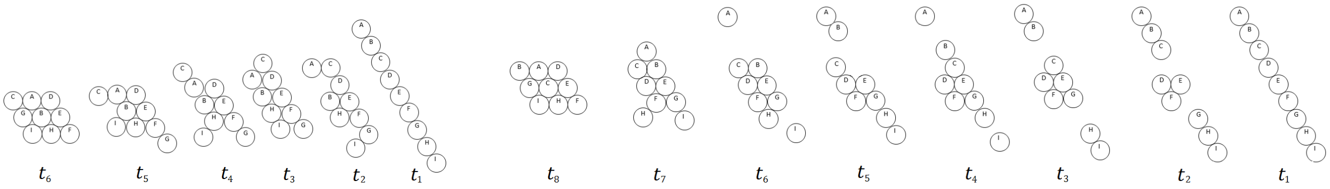


(c) Traveled distance

**Figure 2: Catoms characteristics and features [7]**

The main design goal of Claytronics is to scale to millions of micro-scale units. Each Claytronics MEMS microrobot is equipped with structures called "features". Those features are used as attachments using electromagnetic or electrostatic force and as a direct communication mean. By actuating their features, catoms are able to move around their direct neighbors in the same way stepper motors navigate their surroundings (Fig. 1 and Fig. 2(a)). However, catoms maintain their ability to communicate. Different versions of catoms has been built and dimensions go from 8 cubic meters (Giant Helium Catoms) to 4 centimeters (Planar Catoms in Figure 2(b)) and to few millimeters (Millimeters Scale Catoms) [1]. Bigger catoms are used to evaluate forces interaction and physical properties. Table 1 summarizes technical parameters of catoms such as weight, power consumption and manufacturing methods. The adoption of wireless communications for catoms has been investigated in [4]. Results show that wireless communications in a network enhances the way catoms are communicating, improving the communication range and allowing easy way of broadcasting information. Several catoms form an ensemble, which is a dense environment in the communication range. Most control algorithms would improve performances from broadcasting capability, speeding information diffusion process, allowing neighbor discovery and providing ability to communicate with non contiguous groups of catoms. Broadcasted messages can also be used as a form of synchronization, which is very useful when coordinating movements of groups of catoms. In Claytronics energy is provided to each catom from an external power source, and if required can be routed from catom to catom. From a topological point of view, a catoms ensemble can be viewed as a connected undirected graph $G = (V; E)$ modeling the MEMS network. $v \in V$ , is a node belonging to the network and, $e \in E$ is a bidirectional edge representing the communication between two physical neighbors. For each node $v \in V$, the set of neighbors of $v$ is denoted as $N(v) = u; (u; v) \in E$. Each node $v \in V$ knows the set of its neighbors in $G$, $N(v)$. Furthermore, the set $N(v)$ is the unique initial and instantaneous information received by the node. It is assumed that every node systematically updates the set of its neighbors $N(v)$ after a local change. In [7] the following definitions are given:

- *Connectivity*: a graph $G = (V; E)$ is said connected, if $\forall v \in V; \forall u \in V\ \exists$ a tuple $C_{v,u} = (e_v; -; ...; e_{(-,-)}; ... ; e_{(-,u)})$, where $e_{x,y} \in E$ is a edge from $x$ to $y$ and $C_{v,u}$ represents a path from $v$ to $u$.
- *Snap-Connectivity*: Let $T$ be the full-time running of a Distributed Algorithm (DA), and $t_1, t_2, ..., t_n$ are different time instants of DA execution (rounds). Let the dynamic graph $G_t(V_t; E_t)$ the network state at the in-



(a) PASC (b) PAUC

**Figure 3: An example of execution of PASC [7] and PAUC [10] with 9 nodes**

stant $t$. DA is said respecting Snap-Connectivity property if $\forall t_i, i \in 1, ..., n, G_{t_i}(V_{t_i}; E_{t_i})$ maintains the connectivity.

- *Non-Snap-Connectivity*: there is a non-snap-connectivity if the graph that models the network is connected only at the end of the algorithm.

To calculate the highest number of movements that can happen in an ensemble the following scenario is considered. Considering a microrobot model like the one in Fig. 2(c) it is possible to say that a microrobot has done a single movement if the distance between its former position and its new position is exactly twice the radius $D_1 = 2R$. As an example, if the node is in a position at a distance $D_2$ from the former position it has done two movements. A $360^o$ angle can be divided to six equal angles each one has $60^o$, since the perimeter at an angle a is $Pa = \pi Ra/180$ and $P = 2\pi R$ it is straightforward to say that $P_1 = P_2 = P_3 = P_4 = P_5 = P_6$, this means that the node can have without overlapping at most six neighbors and in each movement the node travels $Ra$ (with $a = 60^o$) from $m0$ to $m$. The research aims to develop self-configurable microrobot swarms that can be deployed in a constrained environment. The following assumptions are made: i) Target shape is not mapped; ii) Every node can communicate directly only to its physical neighbor; iii) Every node runs the same algorithm; iv) Every node has a limited sensing range.

The distributed algorithms listed below have been implemented using the declarative language MELD [14] and evaluated using the simulator DPRSim by Carnegie Mellon University. Most of those algorithms aim at shaping a chain of microrobots (worst case scenario) into a square. Indeed, the chain form represents the worst physical topology for many distributed algorithms in terms of fault tolerance, propagation procedures and convergence. Also, a chain of microrobots represents the worst case for message broadcasting complexity (that is $O(n)$). The redeployment into a square organization (an $\sqrt{n} \times \sqrt{n}$ matrix) allows to obtain the best message broadcasting complexity with $O(n)$. Among several other versions in [8] [11] [12] [13], two main algorithms are presented below as an example of different approaches to model complexity in message exchange.

## 2.1 PASC

The main distributed algorithm is Parallel Algorithm with Safe Connectivity (PASC) in [7]. It is used to move catoms from a chain to a square shape. The catom in the middle of the chain is the initiator and other catoms move around it in a parallel fashion. To ensure Snap-Connectivity of the network, only leaves of the tree are allowed to move. It is a very robust algorithm in which each catom can have 10 states. Complexity of the exchanged messages is $O(n)$, $O(n/2)$ for the tree construction and $O(n/2)$ to find an initiator. This algorithm keeps the Connectivity and Snap-Connectivity property of the graph. Figure 3(a) shows an instance of movements of 9 catoms running PASC.

## 2.2 PAUC

The second distributed algorithm is Parallel Algorithm with Unsafe Connectivity (PAUC) in [9]. With respect to PASC, PAUC ensures Snap-Connectivity only at the end of the procedure, under the assumption that a single catom could freely move. This is not true in general but shows the scenario in which some catoms could be parted from the whole ensemble. It has 7 states and its complexity is $O(n/2)$. Another version of PAUC is presented in [10], this time with 9 states and same complexity. Figure shows how PAUC runs on a chain of catoms.

All the algorithms previously presented have been designed to work on touch communication through features and do not really take advantage of a wireless communication paradigm. In fact, as shown in [4] and [5] it is demonstrated that using nano-wireless communications dramatically enhances the performance of the ensemble communication. For example, isolated catoms that cannot reach the main ensemble can be reintegrated using the catom walker algorithm in [15], helping to build a strong fault tolerance.

# 3. NANO-WIRELESS COMMUNICATIONS

To develop nano-wireless communication the antenna have to be reduced down to a few hundreds of nanometers. The main consequence is that the process requires the use of extremely high operating frequencies, which could compromise the feasibility of electromagnetic wireless communications among nano-devices. However, using graphene to fabricate nano-antennas can overcome this limitation. Carbon nanotubes and graphene nano-ribbons can enable the communications at that frequencies. In [16], a remarkable comparison between nano-patch antennas based on graphene nano ribbons and nano-dipole antennas based on carbon nanotubes is given, showing how a graphene nano-antenna $1\mu m$ long can efficiently radiate EM waves in the Terahertz Band ($0.1 - 10.0$ THz). Nano-wireless communication is therefore a new paradigm being developed in the field of nanotechnology. Building nanonetworks means building nano-sized communicating devices that are able to perform simple tasks at the nanoscale. This technology is feasible for environments with very high node densities, in the order of hundreds of nanosensors per square millimeter. A wide range of applications such as advanced health monitoring systems or high-performance distributed nano-computing architectures are made possible thanks to this new technology. Key features of novel plasmonic nano-transceivers and nanoantennas, which operate in the Terahertz Band, bring along the need to develop tailored communication schemes for nanonetworks [2]. In [3] a modulation and channel access scheme for nanonetworks in the Terahertz Band is developed. The technique is based on the transmission of one hundred- femtosecond-long pulses by following an asymmetric On-Off Keying modulation Spread in Time (TS-OOK). TS-OOK technique uses very short pulses in the range of femtoseconds where each pulse transmits a "1" and the absence of pulse transmits a "0" which is called on-off keying. To ensure an efficient medium sharing and reduce interference and noise, each pulse must be transmitted using a period of time much bigger than the duration of the pulse itself. Of course the transmission of "0" has to be preferred to "1" as it does not create any perturbation of the channel and it is also easier to detect. Using a low-weight channel coding can reduce the influence of interference between concurrent transmissions leading to better aggregated bandwidth. Moreover, Rate Division TS-OOK varies time interval between pulses between transmitters to avoid series of collisions between two nodes. Indeed, in TS-OOK the interval between two pulses is a pre-determined fixed value and is a key parameter. If two nodes are emitting at the same time, all their transmissions would result in collisions. Rate Division TS-OOK improves with a medium access control (MAC). PHysical Layer MAC Protocol for Electromagnetic nanonetworks (PHLAME) uses a handshaking protocol which allows multiple receptions, defines the interval length between two pulses and chooses the best channel coding scheme to increase transmission reliability.

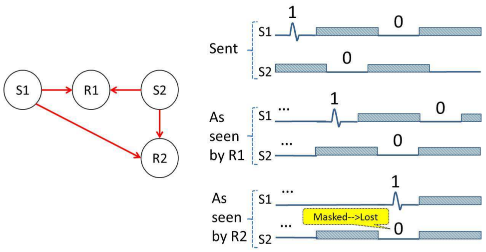


**Figure 4: "0" symbols can be masked by "1" [5] [6]**

This coding scheme is chosen to match a pre-defined packet error rate and consists of two mechanisms. First, the coding is transformed reducing the number of "1"s in favor of the "0"s taking into account that logical "1" will create perturbation on the channel whereas "0"s does not. Furthermore, "0"s can be detected at a further distance. Second, a $n$-repetition code is used where each symbol is replicated $n$-times. The handshaking protocol consists of two phases: a transmission request, starting the handshake and a transmission confirmation, confirming the handshake. The key parameter $\beta$ represents time between two pulses as in Fig.4. The grater the $\beta$, the bigger the multiplexing capability [6]. Pulses are very short (around 100 femtoseconds), the propagation delay (3 nanoseconds per meter) can be greater than the duration of a pulse and collisions can easily occur at receiver side even if symbols were sent at different times.

Figure 4 shows how symbols can overlap from a receiver point of view. Sender S1 sends a "1" (as a pulse) and S2 later sends a "0" as an absence of pulse. Because receiver R1 is at the same distance from S1 and S2, it receives those symbols one after the other and decodes them correctly. Receiver R2 is much closer from S2 than from S1. As the propagation delay from S1 is longer, the two symbols overlap and the "0" from S2 is masked by the "1" from S1. Also, if the nodes are able to move, the propagation delay will not remain constant over time for a given communication link.

However, a node can be affected by a collision only when receiving a "0". If a node is currently receiving a "1", whatever it receives at that time from others nodes, it will still be interpreted as a "1". When receiving a "0", receiving other "0" will not cause a problem. Only concurrent "1" will mask a "0" currently being received, as it would be the case for receiver R2 for the symbol sent by S2 in Fig. 4.

# 4. TOPOLOGY FORMATION VIA NANO-WIRELESS COMMUNICATIONS

As previously mentioned isolated catoms is an unwanted scenario. Under the effect of errors and faults some nodes may disconnect from the main ensemble. The idea is to keep the ensemble as much connected as possible and in case recover from fault situations. A fragmented ensemble can be restored into a connected one as soon as different groups are joint into one. This kind of advanced operations can be performed only using high level protocols, also an addressing scheme is needed. In high node density environments, scalability is also a main matter of concern. For instance, micro and nanorobotics would be able to transmit their information in a multi-hop fashion to a common sink. As result a Wireless Nano Network (WNN) can be modeled so as to apply concepts that are already well-acquainted [17]. A simple addressing protocol could be the one used in Enhanced Hierarchical Routing Protocol (EHRP) implemented in Zigbee low power wireless devices [18].

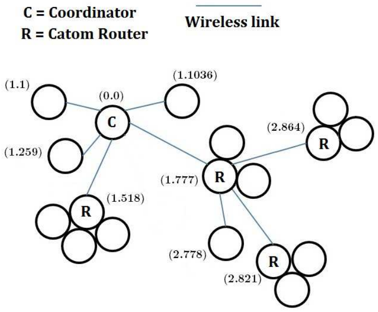


**Figure 5: A possible topology using tree routing addressing scheme**

By using this framework for microrobotics a simple addressing protocol can be designed. In Claytronics, each catom is connected to up to six other catoms in the same ensemble. These six catoms represent the leaves of the routing tree and communication is handled through features assuming no delay. Accordingly, each catom that is enabled to use nanowireless communication forms a tree topology with a coordinator node as the root node. The network addresses are assigned using a hierarchical addressing scheme and each parent gives a unique address to its children. Parameters are defined by the coordinator which determines $C_m$, the maximum number of children a parent can have, $R_m$, the maximum number of routers a parent can have as children, and $L_m$, the maximum depth in the network [19]. With those parameters it is possible to compute $Cskip_d$ which represents the size of the address sub-block being distributed by each parent at depth $d$ to its router-capable child devices.

$$C_{skip_d} = \begin{cases} 0 & R_m = 0 \\ 1 + C_m \cdot (L_m - d - 1) & R_m = 1 \\ \frac{1 + C_m - R_m - C_m \cdot R_m^{L_m - d - 1}}{1 - R_m} & R_m > 1 \end{cases} \qquad (1)$$

Network addresses $A_{d+1,rn}$ and $A_{e+1,el}$ shall be sequentially assigned to the $n$-th router child and $l$-th end device child at depth $d$+1 respectively, according to the following equations:

$$A_{d+1,rn} = A_{parent} + C_{skip_d} \cdot (n-1) + 1, \tag{2}$$

$$A_{d+1,el} = A_{parent} + C_{skip_d} \cdot R_m + l, \tag{3}$$

where $1 < n < R_m$, and $A_{parent}$ represents the address of the parent and $1 \leq l \leq (C_m - R_m)$. This addressing scheme is particularly effective because it fits the constraint of 6 catoms at the bottom of the tree. This addressing approach has been used in the past literature for routing in low power wireless devices exposed in [20]. As for catoms, it allows not only to recover from fragmented ensemble but also to implement fault tolerance policies.

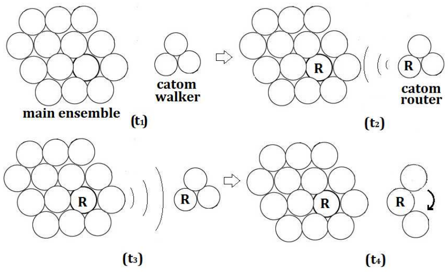


**Figure 6: Catom walker communicating with main ensemble**

While in PAUC it has been assumed that isolated catoms are able to move independently (to minimizes message exchange complexity), in real applications, a single catom may not be able to move freely (or to move at all). The minimum number of catoms required to move is 2, although three catoms have a better mobility. Three catoms which coordinate themselves to independently move are called Catom Walker (CW) [15]. To enable an efficient nanowireless communication, an ensemble elects a catom Router (R). It is assumed that all catoms have nanowireless communication capability, but only catom routers are allowed to use it. It is not strictly required for all catoms to have an address. In fact, the main task of catom router is to handle local communication within its cluster and communicate with other catom routers. Catom routers are particularly effective when there are several different sized ensembles that have to communicate and coordinate together. Catom routers can be defined also within the same ensemble when feature communication gives poor scalability performances. The tree topology does not prevent a catom router child to be itself a router. Fig. 5 shows a possible network topology using the presented addressing scheme. Some of the nodes are connected through features and some through nanowireless. Let's consider two scenarios. In the first one we have a CW that needs to be integrated in the main ensemble. Using only feature communication, such task would have not been possible. Thanks to nanowireless communication an handshake protocol can be implemented so as to allow the CW to move towards the main ensemble. Fig. 6 shows a possible execution of such protocol. At $t_1$ we have that a CW is not integrated in the main ensemble. At $t_2$ the R from the CW periodically broadcasts a message searching for larger ensembles. At $t_3$ the catom router from the main ensemble responds to the request and at $t_4$ the CW starts to move towards the main ensemble. Intuitively, the smaller ensemble move towards the bigger, in order to reduce delay. Optimized distributed algorithms to implement migration of random sized ensembles of catoms are therefore an important issue.

The second scenario is a faulty one, where a single catom from a CW is not working and the overall mobility is affected. In Figure 6 at $t_1$ the faulty CW sends a request for help to near ensembles. At $t_2$ the main ensemble designates a CW to rescue the isolated catoms. At $t_3$ and $t_4$ the designated CW starts moving to rescue the faulty one. The two remaining catom walkers will be integrated into a five-node catom walker and successively integrated in the main ensemble.

**Algorithm 1** Rejoin Algorithm

```
procedure REJOIN($c_i, c_j$)
    while true do
        sendMsg(helloMsg($a_i, p_i, n_i, t_i$))
        checkMailbox()
        if checkMailbox() = newMsg() then
            if newMsg()=helloMsg($a_j, p_j, n_j, t_j$) then
                if $n_j > n_i$ then
                    sendMsg(reqMsg($a_i, p_i, n_i, t_i$));
                end if
                if newMsg()=reqMsg($a_j, p_j, n_j, t_j$) then
                    sendMsg(perMsg(ai, pi, ni, ti))
                end if
                if newMsg()=perMsg($a_j, p_j, n_j, t_j$) then
                    moveToPosition($p_j$)
                    connectFeatures()
                end if
                if newMsg()=helpMsg($a_j, p_j, n_j, t_j$) then
                    $rescuer$ ←selectClosestCatom($p_j$)
                    enableRouting($rescuer$)
                    sendCatomWalker($rescuer$, $p_j$)
                end if
            end if
        end if
        $n_i$ ← checkConnectivity()
        if ($n_i \leq 2$) then
            sendMessage(helpMessage($a_i, p_i, n_i, t_i$));
        end if
    end while
end procedure
```

To approach this problem, a Rejoin Algorithm is presented. Assuming to have two catom routers $c_i$ and $c_j$, and for each of them we know the address $a_i, a_j$ , the position $p_i, p_j$ and the number of catoms in their cluster $n_i, n_j$. At each iteration, a catom router sends a message with timestamp $t_i, t_j$ to scan the environment for other clusters and then checks whether a new message has been received. Several kind of messages can be exchanged: *helloMsg*$(a_j, p_j, n_j, t_j)$ is the basic message sent by catoms to notify their presence to other catoms in the surroundings. *reqMsg*$(a_j, p_j, n_j, t_j)$ is sent by a router who has received and *helloMsg* by a router that belongs to a bigger cluster. *perMsg*$(a_j, p_j, n_j, t_j)$ is a permission to join message sent when a *reqMsg* is received. When a router receives a *perMsg* it starts to move toward the position of the bigger cluster. The functions *moveToPosition*$(p_j)$ and *connectFeature()* are primitives to connect the two clusters through their features. At each iteration a catom router verifies its connectivity and updates its number of catoms connected through features. If the number of catoms is smaller than 3, the a *helpMsg* is sent. When a router receives a *helpMss*, it designates a catom walker and enable the closest catom *rescuer* to use nanowireless com-

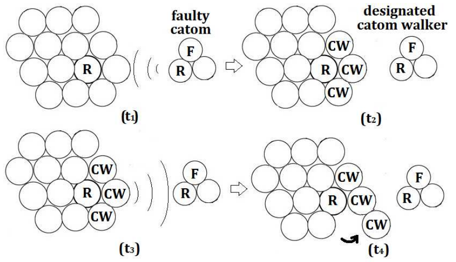


**Figure 7: Faulty catom walker requesting to be integrated in the main ensemble**

munication. Then the *rescuer* catom will be sent to join the faulty catom cluster.

This protocol prevents Claytronics to have a single point of failure and to adopt fault tolerance policies. Routing protocols based on nanowireless communication will result in a wider span of tasks that can be accomplished through catoms cooperations and pave the way to advanced microrobotics applications.

## 5. CONCLUSIONS

This paper presented the possible use of nanowireless communication for microrobotics with emphasis on the Claytronics ones. To improve scalability and to recover possible faults, nanowireless communications using femtosecond-long pulses by following an asymmetric TS-OOK modulation were discussed. A simple addressing scheme for catoms as well as an algorithm to rejoin different catom clusters are presented. Future work will be dedicated to analyse the behaviour and the scalability of such protocols in simulated large microrobots networks.

## 6. REFERENCES

[1] S. C. Goldstein, T. C. Mowry, "Claytronics: A scalable basis for future robots", In RoboSphere 2004, Moffett Field, CA November, 2004.

[2] J. M. Jornet and I. F. Akyildiz "Channel modeling and capacityanalysis for electromagnetic wireless nanonetworks in the terahertz band", IEEE Trans. Wireless Commun., vol. 10, no. 10, pp. 3211-3221, 2011.

[3] J. M. Jornet and I. F. Akyildiz, "Femtosecond-long Pulse-based Modulation for Terahertz Band Communication in Nanonetworks," IEEE Transactions on Communications, vol. 62, no. 5, pp. 1742-1754, May 2014.

[4] N. Boillot, D. Dhoutaut, and J. Bourgeois. "Efficient Simulation Environment of Wireless Radio Communications in MEMS Modular Robots." In iThings 2013, IEEE Int. Conf. on Internet of Things, Beijing, China, pages 638–645, August 2013.

[5] N. Boillot, D. Dhoutaut, J. Bourgeois, "Using Nano-wireless Communications in Micro-robots Applications", In NANOCOM 2014, 1st ACM Int. Conf. on Nanoscale Computing and Communication, pages 1-9, Atlanta, Georgia, USA, May, 2014

[6] N. Boillot, D. Dhoutaut, J. Bourgeois, "Parameter study and characterization of wireless nanonetworks through simulation", Communications and Networking (BlackSeaCom), 2014 IEEE International Black Sea Conference on (pp. 43-47). IEEE, May 2014.

[7] H. Lakhlef, H. Mabed, J. Bourgeois: "Parallel self-reconfiguration for MEMS microrobots". EUROCON 2013: pp.283-290, July 2013.

[8] H. Lakhlef, H. Mabed, J. Bourgeois, "Distributed and efficient algorithm for self-reconfiguration of MEMS microrobots", In SAC 2013, 28-th ACM Symposium On Applied Computing, pages 1-6, Coimbra, Portugal, March, 2013.

[9] H. Lakhlef, H. Mabed, J. Bourgeois "Efficient Parallel Self-Reconfiguration Algorithm for MEMS Microrobots", PDP 2014: 154-161, Feb 2014.

[10] H. Lakhlef, J. Bourgeois, H. Mabed, "Robust Parallel Redeployment Algorithm for MEMS Microrobots". AINA 2014: 1057-1064, May 2014.

[11] H. Lakhlef, H. Mabed, J. Bourgeois, "Optimization of the logical topology for mobile MEMS networks", J. Network and Computer Applications 42: 163-177, Jun 2014.

[12] H. Lakhlef, H. Mabed, J. Bourgeois, "An energy and memory-efficient distributed self-reconfiguration for modular sensor/robot networks. The Journal of Supercomputing 69(2): 908-929, Aug 2014.

[13] H. Lakhlef, J. Bourgeois, H. Mabed, S. C. Goldstein, "Energy-aware parallel self-reconfiguration for chains microrobot networks". J. Parallel Distrib. Comput. 75: 67-80 Feb 2015.

[14] Ashley-Rollman, M. P., Goldstein, S. C., Lee, P., Mowry, T. C., Pillai, P. (2007b) "Meld: A Declarative Approach to Programming Ensembles". In Proceedings of the IEEE International Conference on Intelligent Robots and Systems IROS '07.

[15] N. Boillot, D. Dhoutaut, J. Bourgeois, "Large scale MEMS robots cooperative map building based on realistic simulation of nano-wireless communications", Nano Communication Networks, Feb 2015.

[16] J.M. Jornet, I.F. Akyildiz, "Graphene-based nano-antennas for electromagnetic nanocommunications in the terahertz band", in: Proc. of 4th European Conference on Antennas and Propagation, EUCAP, April 2010.

[17] J.M. Jornet, I.F. Akyildiz, "Electromagnetic wireless sensors network", Nano Communication Networks Volume 1, Issue 1, Pages 3-19, March 2010.

[18] J. Y. Ha, H. S. Park, S. Choi, W. H. Kwon, "EHRP: Enhanced hierarchical routing protocol for zigbee mesh networks", Communications Letters, IEEE, Volume:11 , Issue: 12, pp. 1028 - 1030, Dec 2007.

[19] P. Ran, M. Sun, and Y. Zou, "ZigBee routing selection strategy based on data services and energy-balanced ZigBee routing," in Proc. IEEE Asia-Pacific Conf. on Services Computing 2006, pp. 400-404, Dec. 2006.

[20] F. Cuomo, S. D. Luna, U. Monaco and T. Melodia, "Routing in ZigBee: benefits from exploiting the IEEE 802.15.4 association tree", IEEE International Conference on Communications, pp. 3271 - 3276, June 2007.

[21] Rubenstein, M. Ahler, C., Nagpal, R., "Kilobot: A low cost scalable robot system for collective behaviors", 21st Mediterranean Conference on Control & Automation (MED), 2013, pp. 1 - 6, June 2013.