



HAL
open science

New communicating concrete for data storage and retrieval through integrated micro sensor nodes

Kais Mekki, William Derigent, Ahmed Zouinkhi, Eric Rondeau, André Thomas, Mohamed Naceur Abdelkrim

► To cite this version:

Kais Mekki, William Derigent, Ahmed Zouinkhi, Eric Rondeau, André Thomas, et al.. New communicating concrete for data storage and retrieval through integrated micro sensor nodes. 4th International Conference on Future Internet of Things and Cloud, FiCloud 2016, Aug 2016, Vienne, Austria. hal-01356595

HAL Id: hal-01356595

<https://hal.science/hal-01356595v1>

Submitted on 25 Aug 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

New Communicating Concrete for Data Storage and Retrieval through Integrated Micro Sensor Nodes

Kais Mekki^{1,2}, William Derigent¹, Ahmed Zouinkhi², Eric Rondeau¹, André Thomas¹, Mohamed Naceur Abdelkrim²

¹Université de Lorraine, CRAN, UMR 7039, 2, avenue de la forêt de Haye, Vandoeuvre-lès-Nancy Cedex, 54516, France.

²Research Unit of Modeling, Analysis and Control of Systems (MACS), National Engineering School of Gabes, rue Omar Ibn Elkhattab, 6029 Gabes, Tunisia.

{kais.mekki,william.derigent,eric.rondeau,andre.thomas}@univ-lorraine.fr
{ahmed.zouinkhi,naceur.abdelkrim}@enig.rnu.tn

Abstract—A new area is coming with intelligent materials able to provide diverse functionalities to users all along the product lifecycle, during the design, manufacturing, use and dismantling phases. These materials can track their own evolution all along the product lifetime, gather helpful information and thus allow information continuum at all time and everywhere. Usually, these functionalities are fulfilled via the integration of specific electronic components into the material (wireless sensors nodes, RFID tags). The present paper forms part of this framework in considering that thousands of micro-sensor nodes are integrated into a concrete precast. All nodes store different information of precast properties in their memory requiring designing a specific Wireless Sensor Network (WSN) protocol to extract this information. The proposed protocol employs three mechanisms. First, the random walk is used as multi-hop process to forward the request to further nodes. Then, the pull gossip mechanism is added to interrogate the neighbor nodes in each hop. Finally, a mechanism called farthest neighbors selection is added. Only the farthest and most powerful neighbor is selected as next hop for improving the reliability of the original random walk process.

Keywords— *Smart homes, Communicating materials, Precast concrete, Micro sensor networks, Data retrieval protocols.*

I. INTRODUCTION

Concrete's versatility, durability, and economy have made it the world's most used construction material. The concrete construction companies mainly follow two strategies: i) Local concrete fabrication: the concrete is produced on construction site via formwork, the site is delivered only in raw material (granula, cement, adjuvant, etc.) and scrap. ii) Precast concrete fabrication: all construction elements (beams, floors, bearing walls, pillars, etc.) are factory away from the site and brought to the site to be implemented quickly.

For reasons of cost and time, the choice is take up mostly toward the precast concrete. Indeed, as it was described in one of the four roles of construction management [1]: *We must transfer the activities of the earlier building site in the chain.* These avoid the climate change in the site and conduct a number of parallel operations. With precast, it is possible to improve the construction site in place, time and quality. Actually as example, the United States uses about 260 million

cubic meters (340 million cubic yards) of ready precast concrete each year. It is used in highways, streets, parking lots, parking garages, bridges, high-rise buildings, dams, homes, floors, sidewalks, driveways, and numerous other applications [2].

In recent years, a new research area has appeared for improving concrete in logistic chain, traceability and security using Internet of Thing tools (especially RFID and Sensor Nodes technology). As example in 2007, BizzDev company (www.bizzdev.com) designed a system composed of two parts: a temperature sensors node is directly embedded in the precast and RFID tag in its surface. The concrete is therefore able to deliver its temperature continuously over time for supervising application. Also, the CERIB company (www.cerib.com) on collaboration with CRAN laboratory uses RFID in the concrete for traceability in the supply chain: concrete beams were instrumented by integrating two RFID tag on each end. Then, CE information has been stored in the memory of these tags, increasing the chances that such information is not lost and still available throughout the supply chain.

However, the idea used in this two examples consists in integrating sensor nodes and tags RFID in some part of the concrete precast. This could lead to two problems: i) If the precast exceeds the human scale. It is common to find some buildings beams or concrete slabs of tens meters (even a hundred meters). It is then difficult for the operator to systematically go through all of the building (for example, over 100m) to identify its characteristics or by searching for the tags containing the information. ii) When the concrete precast is used and installed inside the building, it is difficult to localize and access to the RFID tags inside the precast. iii) The RFID technologies are limited in memory, thus using one or two tags could not allow large storage memory of information.

For this reason, another's application aims to fully integrate RFID tags in concrete precast to allow accessing information from each part of building. In [3], the RFID tags are placed uniformly in the surface of the precast to take a first step towards automatic detection of cracks. The cracks are detected

when some tags are damaged. In [4], Lafarge company (www.lafarge.com) integrated RFID tags directly into the concrete for traceability application. The RFID tags are disseminated in all the building concrete at number 4 or 5 tags every 2m^3 on average. The ID of tags could then be read in each part of building. Using the ID, the operators could access to all the needed information about the concrete (e.g. constructor, date, technical characteristic, etc.) through external associated database.

The nature of the two last above application is limited. It is a simple identification of RFID tags and a consultation of concrete data from an external database.

In this paper, we define new services of communicating concrete precast which could store database information directly into the concrete. This paradigm is developed through uniformly integrating thousands of micro-sensor nodes in the concrete (25 nodes each 4m^2). Each node could communicate wirelessly with others inside the concrete. Thus, the nodes could then construct a wireless sensor network inside the building. Then, the information could be stored in the micro-nodes using data storage protocol for wireless sensor network. Unlike RFID, the operator could connect to any node in the concrete and store or read all the information in the building through multi-hops routing protocol of WSN.

For this issue, we recently developed a new data storage protocol called USEE [5]. USEE guarantees that each data is uniformly replicated throughout the WSN inside the precast. Thus, information could be read in all parts of the building. USEE is based on three strategies: counter-based flooding to forward messages to all nodes in the concrete, neighbor storage information which control the replication rate and ensure that information is stored once in each neighborhood as much as possible (i.e. for increasing the storage capabilities of WSN), and finally probabilistic storage (i.e. the data is stored with probability P) [5]. This protocol was showed good performance in dense sensor network through simulation especially for uniform data storage in the whole network, optimization of energy consumption and high storage capacity compared to the literature protocols. To extract and read the stored data, this paper presents a data retrieval protocol based on Random Walk (RW). The RW empowers nodes with the ability to efficiently forward requests hop by hop which only rely on neighborhood information. We empowered RW by adding Pull Gossip (PG) to query the neighbors in each hop and increase the number of covered nodes during retrieval process.

In the rest of this paper, the new data retrieval protocol is firstly detailed. Then, the new paradigm of communicating concrete precast using micro-sensor networks is presented. We studied the optimal density of the nodes inside the concrete precast (i.e. probabilistic and simulation studies) for the best uniform storage results of USEE protocol and for the best reliability of the proposed data retrieval protocol. Finally, the retrieval protocol performances are discussed in terms of number of covered nodes, response delivery ratio, and average delay.

II. DATA RETRIEVAL PROTOCOL IN COMMUNICATING CONCRETE

In this section we propose a data retrieval protocol to read the uniformly replicated data in the concrete by USEE [5]. In the following, a state of art of data retrieval protocol of WSN is presented. Then, the proposed data retrieval protocol is detailed.

A. State of art

Various schemes to efficiently retrieve data in WSN have been proposed in the past years [6]. There are two main approaches: *two request/response phases* and *one request/response phase*.

- *Two request/response phases*: using this class, the user (or sink node) broadcasts a request message, for instance, “give me the temperature in a particular area”. The request is flooded throughout the network. Upon receiving the request, each node sets up a gradient toward the nodes from which it receives this message (i.e. the gradient specifies both data rate and the direction along which response should be sent). This process continues until gradients are set up from the sources back to the sink (figure 1(a)). The source data are then returned in the reverse path of the request propagation. If each node receives the same request from more than one neighbor, the source data will travel to the sink along multiple paths (figure 1(b)). At this point the sink reinforces a particular path with the least-delay (figure 1(c)). All response messages then could be sent from the source to the sink through the reinforced path (figure 1(d)). Directed Diffusion (DD) [7] is the most known protocol of this class. Various other based DD protocols are developed such as DCHT [8], BMR [9], EAR [10], and GBR [11].

- *One request/response phase*: in WSN, minimizing the number of flooding reduces the energy consumption rate. One request/response class avoids the two existing floods in figure 1(b) and 1(c). First, the request is flooded in the network and gradients are established at each hop. Unlike two phases protocols, when the request reaches a source node, it sends the response messages only to the most powerful neighbor (evaluated thanks to properties like latency, quality of radio link, high residual energy, etc.) in the reverse path of the request propagation. The most known one request/response phase protocols are: MARP [12], MCFA [13], GRAB [14], RR [15], EEQR [16], EQSR [17], and MR2 [18].

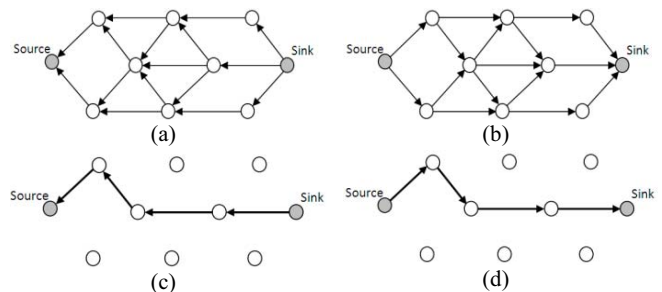


Fig. 1. Two request/response phases for data retrieval in WSN.

In this paper, choosing the correct data retrieval protocol is related to the specificities of the communicating concrete and

the way the data is stored inside it by USEE [5]. The information is uniformly replicated in the network inside the concrete (the same information is replicated in many nodes). However, all the literature techniques are based on request flooding. By using flooding technique in communicating concrete, many nodes could respond to the same request leading to high redundant rate, network congestion, high transmission rate, and thus increasing the energy consumption. Hence, using of such protocols is not appropriated since they deplete the energy of nodes especially for limited resources of micro-sensor node.

Thus, Random Walk (RW) [19] is chosen in this paper to propagate the request hop by hop throughout the WSN. However, this mechanism could lead to random long path (data retrieval failure) and long delay. Thus, Pull Gossip (PG) [20] mechanism is also selected to interrogate the neighbor nodes in each hop of the RW process.

B. Random Walk and Pull Gossip based protocol for data retrieval in communicating concrete

The Random Walk (RW) is a process in which a carried data is repeatedly forwarded (as a message) from a node to one of its randomly chosen neighbors until the end of the random walk length (i.e. the number of hops that the random walk should last) as shown in figure 2(a).

As discussed before, the RW process could lead to random path, and could lead to long path without any results. Thus, the Pull Gossip (PG) mechanism is executed in each hop. Using PG based algorithm, a node query its neighbors. The node will receive a response only if a neighbor has this information. Thus, the request message is transmitted from a node X to another Y only if the information is missing in the neighborhood of X . This process increases the number of covered nodes of simple RW as shown in figure 2(b).

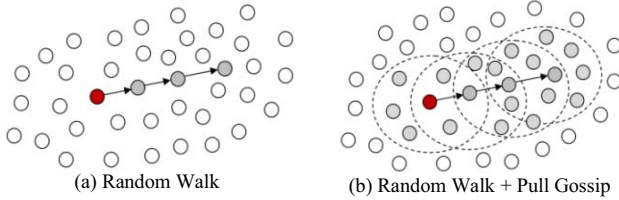


Fig. 2. Random Walk and Pull Gossip mechanisms.

The next hop is randomly chosen. However, the information has more chance to be found if many new nodes are covered in each hop, to increase the efficiency of the retrieval process. Thus, this process is modified by forwarding the request to the farthest neighbor node, hence the maximum number of nodes in the next PG discover is covered. As example, in figure 3(a), the next hop is chosen randomly. The number of new nodes covered by PG during the next hop is equal to 6. However, in figure 3(b), the next hop is chosen thanks to farthest neighbor selection. The number of new covered nodes is equal to 12. This mechanism dramatically increases the chance of finding information at each hop. The farthest neighbors selection algorithm is described in the next section.

Moreover, a link cost function is used by the node to select the best farthest neighbor as next hop. Thus, the request is routed throughout a high quality of service path which increase the reliability of the protocol. A cost function is used such as presented in [21]. Let L_{Border} is the set of farthest neighbors of a node X inside the concrete. The cost function includes an energy factor, available buffer factor, and interference factor with appropriate weights α , β , and γ :

$$Next\ hop = \max_{Y \in L_{Border}} \{ \alpha E_{resd,Y} + \beta B_{buffer,Y} + \gamma I_{interference,XY} \}$$

Where, $E_{resd,Y}$ is the current residual energy of a farthest neighbor Y where $Y \in L_{Border}$, $B_{buffer,Y}$ is the available buffer size of node Y , and $I_{interference,XY}$ is the Signal-to-Noise Ratio (SNR) for the link between X and Y .

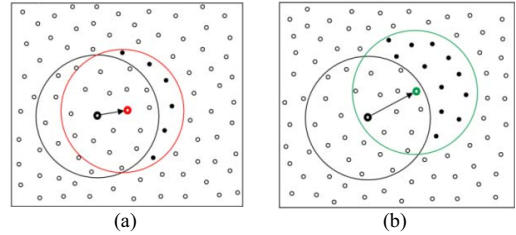


Fig. 3. Benefit of selecting the farthest neighbor in RW process.

C. Farthest neighbors selection algorithm

The algorithm of farthest neighbors selection is executed in the beginning of the concrete precast lifetime. It is used for executing the final data retrieval protocol during the precast lifecycle. The algorithm of farthest neighbor selection uses three messages:

- HELLO: It allows each node to announce their ID to its neighbor nodes.
- TAB_EXCHANGE: It allows each node to announce its neighborhood table to all its neighbors.
- FAR_NEIGHBORS: It allows each node to announce the list of its selected farthest neighbors.

The algorithm of the farthest neighbor selection is executed by all nodes in the concrete. Let X be a node inside the communicating concrete. To construct the neighbor table (N_{Table}) for each node, the HELLO message is exchanged between neighbor nodes using the broadcast communication model. Then, each node inserts its N_{Table} into a TAB_EXCHANGE message and broadcast it to all neighbors. Hence, a node X can compute the number of common nodes between its N_{Table} and the N_{Table} of its neighbors (N_{Table} exchanged using TAB_EXCHANGE). A node Y is considered as farthest neighbor for X , if it has a minimum number of common nodes between N_{Table} of X and N_{Table} of Y .

Then, the list of selected farthest neighbors (L_{Border}) is broadcasted using FAR_NEIGHBORS message. When a node X receives FAR_NEIGHBORS from its neighbor Y , X checks if its ID exists in the list L_{Border} carried by the message. If yes, the node X store the ID of Y (ID_SOURCE of FAR_NEIGHBORS). Finally, X is configured as farthest neighbor for Y .

Once the farthest neighbors selection algorithm is finished, the data retrieval protocol can be executed during the communicating concrete precast lifecycle.

D. Data retrieval protocol

The proposed data retrieval protocol uses five messages:

- REQUEST: It is the request message that contains the requested data reference. It contains the TTL counter which fixes the maximal number of hops for data retrieval path.
- REQUEST_NG: It is the request message that is used to interrogate the neighbors through broadcast communication model.
- FAR_INFO: It is sent by the farthest neighbor toward the REQUEST_NG source. It contains the information needed for the link cost function to choose the most powerful next hop.
- RESPONSE: It is the response message sent by the node that has the requested data in its memory. RESPONSE is sent to the source node (the node that begins REQUEST transmission in the concrete) through the reverse path of the REQUEST message.
- FAILURE: It is sent by the node that receives REQUEST message with TTL=0. FAILURE is sent to the source node through the reverse path of the REQUEST message.

The proposed protocol includes three treatments: REQUEST treatment, REQUEST_NG treatment, and RESPONSE/FAILURE messages treatment. In the following, the treatment of each message is detailed.

1. REQUEST message treatment

The REQUEST message is forwarded from one node to another one in the concrete until a node that contains the information is founded or the TTL counter falls to zero. If one of the latter happens, a RESPONSE or FAILURE message is sent toward the source node on the reverse path of REQUEST. For that, upon receiving REQUEST message, the node temporarily stores the ID of the sender node (ID_SOURCE) to construct the reverse path to the source node. Then, the node checks whether it has the requested information in its memory. Otherwise, the node checks the TTL field in REQUEST. If TTL is zero, a FAILURE message is sent back. Otherwise, the node starts the neighborhood interrogation. For this, a REQUEST_NG message is broadcasted to all neighbors, and then the node waits for reception of FAR_INFO of the farthest neighbors and a possible RESPONSE message. If a RESPONSE is received from a neighbor, the node sends it to the previous node from who it was received the REQUEST. If any RESPONSE message is received, it means that the information does not exist in the neighborhood. In that case, the node decrements TTL, and forwards REQUEST to the best farthest neighbor who is selected using the received FAR_INFO messages and the link cost function.

2. REQUEST_NG message treatment

Upon receiving a REQUEST message, a node X broadcasts REQUEST_NG to all its neighbors. When a neighbor node receives the REQUEST_NG message, it sends back a RESPONSE message to REQUEST_NG sender only if it has

the requested data in its memory. Only the neighbors which are configured as farthest nodes for X send FAR_INFO. Otherwise, the REQUEST_NG message is dropped. As described before, FAR_INFO contains the needed information for the link cost function to evaluate the best powerful farthest neighbor.

3. RESPONSE/FAILURE message treatment

Upon reception of a RESPONSE or FAILURE message, the node sends the message to the already stored ID_SOURCE of REQUEST. Thus, these messages are forwarded through the reverse path of REQUEST until it reaches the source node.

III. INSTRUMENTATION OF THE COMMUNICATING CONCRETE PRECAST

This section details the ideal type of micro-nodes for the instrumentation of the communicating concrete precast. We also study the optimal density of nodes in the concrete for the better performance of USEE protocol in term of uniform replication, through probabilistic and simulation studies. Finally, we study the density of nodes in the concrete for the best performance of the farthest neighbors selection algorithm used by the data retrieval protocol.

A. Micro-sensor nodes for communicating concrete

The proposed communicating concrete paradigm uses micro-sensor nodes which are very small and could be integrated in a precast without modifying either its properties or its appearance as shown in figure 4 (i.e. a uniform nodes deployment is used in the concrete). Currently, several researches focus on miniaturization of electronic components for sensor nodes. But few micro-nodes exist in the literature such as μ Part [22], ECO [23], ZNI [24], SAND [25], SensorCube [26], and Tyndall [27].

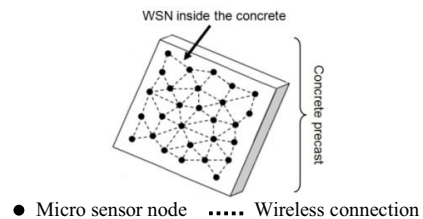


Fig. 4. Communicating concrete precast using micro-nodes.

SAND, μ Part, and Tyndall are the micro nodes with the smallest size (1 cm^3). However, μ Part node has no reception functionality (only transmission data node) and does not allow multi-hop communication. In addition, Tyndall and SAND are very limited lifetime nodes (in average 10 hours and 432 hours, respectively) and require continuous battery charging which seems difficult when the node is embedded in the concrete. However, the SensorCube node has several advantages compared to ZNI and ECO.

First, SensorCube provides temperature and humidity sensors required by the communicating concrete application. Moreover, it provides good memory size as a micro node (120 Kb). Also, it has the advantage of providing a packaged

system being fully reconfigurable. Each module has a specific functionality, such as transceiver, micro-controller, power supply or sensor. In addition, *SensorCube* node could be equipped with many different energy resources such as battery and energy scavenging module, which would increase its autonomy.

For all these benefits, *SensorCube* node was chosen for communicating concrete. In this paper, this node is implemented in the Castalia simulator.

B. Nodes density inside the concrete for uniform data storage using USEE

In this section, the ideal density of nodes inside the concrete precast (nodes are uniformly deployed inside the concrete) is studied for the best performance of uniform data storage using USEE protocol. For this issue, we suppose that: if the data is stored at least once in each neighborhood in the WSN, so the data certainly exist throughout all the concrete precast. In the following, a probabilistic and simulation studies are presented.

a) Probabilistic study

Let N the total number of nodes in the concrete. After executing USEE protocol, the sensor network contains two types of nodes: nodes that store the data and nodes that did not store it. Let:

e_1 : node containing the data

e_2 : node empty of data

E_1 : {all nodes e_1 }

E_2 : {all nodes e_2 }

As presented before, the storage probability of USEE is equal to P [5], let:

$R = \text{Card}(E_1) = NxP$: total number of nodes e_1

$N-R = \text{Card}(E_2) = Nx(1-P)$: total number of nodes e_2

Let n the number of nodes in the neighborhood of a sensor node in the concrete. The n nodes are composed of nodes e_1 and e_2 . In statistics, the neighborhood of size n is a drawing of n nodes among the total number of nodes N . Let attach now to e_1 and e_2 , a number X taking the value 1 if e_1 is drawn and the value 0 if e_2 is drawn. So, X is a discrete random variable representing a *Bernoulli* variable whose probability distribution is given by the two probabilities $\text{prob}(X = 1) = P$ and $\text{prob}(X = 0) = (1-P)$.

We are interested in the random variable k defined as "the number of nodes e_1 among the n nodes". Let, $p(k)$ is the probability of finding k times the e_1 among n nodes.

In our case, the draw of n nodes from N is a simultaneous drawing (not orderly and without replacement). Thus, the probability distribution associated with this draw is a *hypergeometric* probability law with associated parameters n , N and P . It is defined by the following equation:

$$p(k) = \frac{C_R^k \times C_{N-R}^{n-k}}{C_N^n} = \frac{C_{nP}^k \times C_{n(1-p)}^{n-k}}{C_N^n}$$

We are interested in calculating the probability $p(k \geq 1)$ defined as the finding data in the neighborhood at least once. This corresponds to the probability of finding a node e_1 at least

once among the n nodes. The probability $p(k)$ follows a *hypergeometric* law, so:

$$\sum_{i=0}^n p(k = x_i) = 1$$

Where: $x_i = \{0,1,2,3, \dots, n\}$

This gives:

$$p(k=0) + \underbrace{p(k=1) + p(k=2) + \dots + p(k=n)}_{p(k \geq 1)} = 1$$

$$p(k=0) + p(k \geq 1) = 1$$

So, $p(k \geq 1)$ is:

$$p(k \geq 1) = 1 - p(k=0)$$

Finally:

$$p(k \geq 1) = 1 - \frac{C_{N-R}^n}{C_N^n}$$

Figure 5 shows the variation of $p(k \geq 1)$ for different sizes of the neighborhood n and for each storage probability P . the figure shows that the greater the number of nodes n in each neighborhood, the probability of finding the information in each neighborhood increases. This probabilistic study shows that beyond a neighborhood size of 40 nodes, the information exists in every neighborhood regardless of the probability storage P (see figure 5).

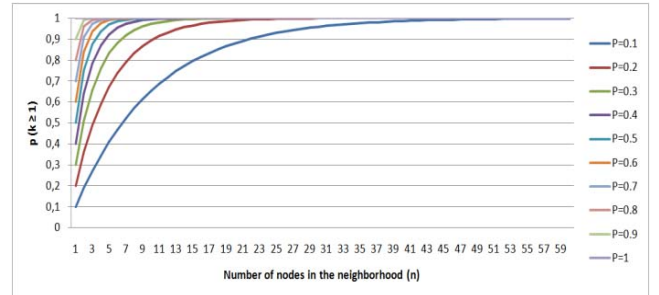


Fig. 5. Variation of $p(k \geq 1)$ for different sizes of the neighborhood n and for each storage probability P .

b) Simulation study

In this simulation study, we validate the probabilistic results obtained in the previous section. For this, we simulate a communicating concrete precast using Castalia simulator. All the nodes are uniformly deployed within a 20x20m square. Wireless radio channel characteristics such as signal noise, interference ratio, and average path loss are chosen to simulate the realistic modeled radio wireless channel in Castalia based on lognormal shadowing and the additive interference models. Different number n of nodes in the neighborhood are simulated: 9, 15, 25, 37, and 45.

Figure 6 shows the data existence ratio in the neighborhood for different sizes of n and for each storage probability P . The figure shows that, when the number of neighbor nodes n is increased, the probability of data existing in the neighborhood increases which gives a probability of more and more to find information in every piece of the precast. Following this

figure, the disseminated data is present in the all neighborhood of almost all the nodes of the concrete (more than 80% and 90% of neighborhoods contain the data) for storage probabilities more than 0.2 and for $n = 25$, $n = 37$ and $n = 45$. However, the data existence probability in the neighborhood is very low for few number of nodes $n = 9$ (between 31% and 62%) and $n = 15$ (between 47% and 85%).

In conclusion, a large number of nodes n in each neighborhood in the sensor network increases the probability of finding data in all parts of the concrete precast.

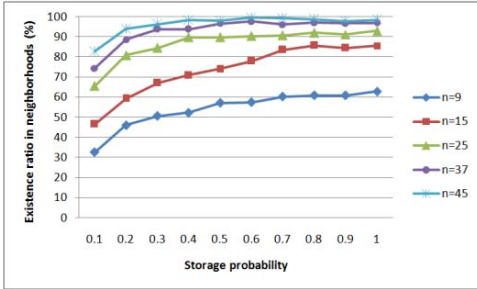


Fig. 6. Data existence probability in neighborhood vs. Storage probability P.

C. Nodes density inside the concrete for efficient farthest neighbors selection algorithm

In this section, we study the impact of the variation of the neighborhood size n on the effectiveness of the farthest neighbor selection algorithm of the proposed data retrieval protocol. The validation of the effectiveness of this algorithm is a key function for the data retrieval protocol.

The red nodes in figure 7 show the selection of the farthest neighbors. The results observation highlights that the proposed algorithm is high efficient for the highest neighborhood density ($n = 25$, $n = 37$, and $n = 45$). The algorithm perfectly works for a communicating concrete using high density of neighborhoods nodes. Low density in the neighborhood does not meet the expected results by the algorithm.

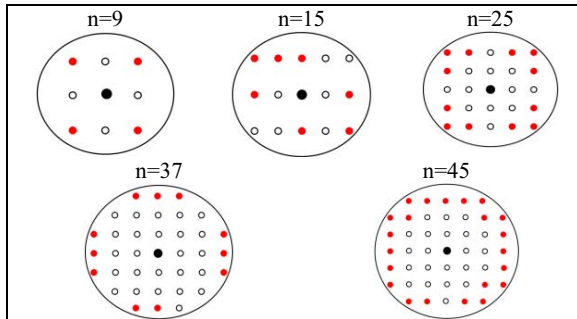


Fig. 7. Effectiveness of the farthest neighbor selection algorithm.

IV. DATA RETRIEVAL PROTOCOL PERFORMANCE

In this section, the data retrieval protocol is evaluated in terms of the number of covered nodes, delivery ratio, and delay metrics. The proposed protocol is compared with the performance of a simple RW+PG algorithm (without farthest neighbor selection and without link cost function).

The proposed protocol uses TTL parameters to limit the retrieval path length. Thus, the maximum length of TTL is

studied through statistics tools using the data existence probability in the neighborhood as parameters (see figure 6). In the following, the TTL value is firstly studied. Then, the simulation results of the protocol performances are presented.

A. Maximum length of TTL

Using USEE, the information is stored in each sensor node with probability P [5]. This process resulted in a probability of existence of information in a neighborhood $p(k)$ as discussed before in this paper. Let p be the probability $p(k)$ of existence of information in a neighborhood.

In the data retrieval protocol, the request is broadcasted in the neighborhood of each hop. Each broadcast, there are two possible cases: i) the neighborhood contains the data in the memory of one of the neighborhood nodes (success), ii) any neighborhood node contains the data (failure). This presents a *Bernoulli* test where each launch has only two possible outcomes, called success and failure.

Therefore, the data retrieval process presents a random experiment of repeating m times (m hops) a *Bernoulli* test of p as parameter. This *Bernoulli* schema is modeled by a tree that had two initial branches (success-failure), and each of these branches gives rise to two new branches (success-failure).

A succession of m independent Bernoulli trials ($m \in \mathbb{N}^*$) allows the construction of a random variable S_m counting the number of success after m Bernoulli trials. This random variable has the *Binomial* probability distribution with parameters m and p . S_m is defined as follows:

$$S_m = \sum_{i=1}^m x_i$$

According to the *Binomial* distribution, for all $0 \leq k \leq m$:

$$P(S_m = k) = \binom{m}{k} p^k q^{m-k}$$

Where :

k : the number of successes.

m : the number of hops.

p : the probability of success in a hop.

q : the probability of failure in a hop.

Let $P(S_m \geq 1)$ the probability of finding the information at least once after m hops:

$$P(S_m \geq 1) = \sum_{k=1}^m P(S_m = k)$$

Moreover, by removing the case of not finding data after m hops:

$$\sum_{k=0}^m P(S_m = k) = P(S_m = 0) + \sum_{k=1}^m P(S_m = k)$$

Using the *Binomial* distribution function of a random variable S_m :

$$\sum_{k=0}^m P(S_m = k) = 1$$

Thus:

$$P(S_m \geq 1) = 1 - P(S_m = 0)$$

$$P(S_m \geq 1) = 1 - \binom{m}{0} p^0 q^m$$

$$P(S_m \geq 1) = 1 - q^m$$

$$P(S_m \geq 1) = 1 - (1 - p)^m$$

$$1 - P(S_m \geq 1) = (1 - p)^m$$

$$\log(1 - P(S_m \geq 1)) = m \times \log(1 - p)$$

For all $P(S_m \geq 1) \neq 1$ and $p \neq 1$:

$$m = \frac{\log(1 - P(S_m \geq 1))}{\log(1 - p)}$$

This result limits the length of the retrieval path. To be almost certain to find the information after m hops, we fixed $P(S_m \geq 1) = 0.99$. Finally, for all $0 < p < 1$:

$$m = \frac{-3}{\log(1 - p)}$$

Figure 8 shows the path length m (which presents TTL value) for different data existence probabilities in neighborhood p . As example, for the probability $0.8 < p < 1$ of $n=45$, TTL should be fixed to 4. For the probability $0.7 < p < 1$ of $n=37$, TTL should be fixed to 5. For the probability $0.6 < p < 1$ of $n=25$, TTL should be fixed to 8.

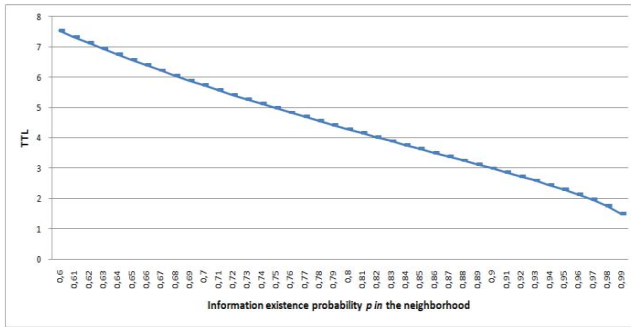


Fig. 8. Probability p vs. TTL.

B. Data retrieval protocol performances

In this section, the performance of the proposed data retrieval protocol is evaluated and compared with the performance of a simple RW+PG algorithm. For this simulation, the neighborhood nodes density $n=25$ is used. For that, 2500 nodes are uniformly deployed in 20×20 m concrete square (25 nodes in each 4 m^2). TTL=8 is used for $n=25$ as presented in the previous section. Finally, many data are stored in the simulated concrete using USEE protocol (one data for each storage probability P). The simulation experience is repeated 60 times and the average results are showed in the following, for the number of covered nodes, delivery ratio, and delay metrics.

a) Number of covered nodes

Figure 9 shows the number of nodes covered using the proposed data retrieval protocol and the simple RW+PG according to the number of hops. RW+PG solution randomly

selects a neighbor node as next hop for request forwarding. This could lead to select neighbor close to the previous hop and hence the new coverage includes many nodes already covered by the previous hops. This issue is avoided by using our farthest neighbor selection technique. Figure 9 indicates the number of the new covered nodes in each next hop is more than RW+PG. The difference between the two protocols is clearer for high number of TTL. For TTL=3 as example, our protocol covers 8 nodes more than RW+PG. However, for TTL=8, the protocol covers 32 nodes more than RW+PG.

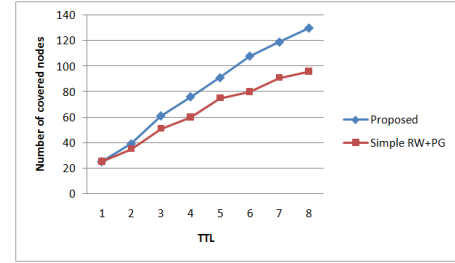


Fig. 9. Number of covered nodes RaWPG vs. simple RW+PG.

b) Response delivery ratio

Another important metric to be assessed is the average response delivery ratio. Figure 10 shows the average delivery ratio of the proposed protocol and the simple RW+PG. Figure 10 highlights that the proposed protocol outperforms RW+PG due to the consideration of the link cost function to choose the most powerful farthest node as next hop. Thus, the request and response messages are forwarded through a high quality of service path. Implementing a link cost technique in the data retrieval algorithm enhances the delivery ratio of the request and the response messages.

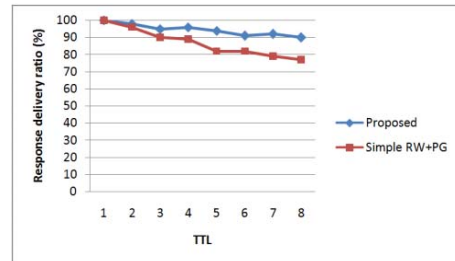


Fig. 10. Average response delivery ratio of RaWPG vs. simple RW+PG.

c) Average delay

End to end delay is also an important metric in evaluating data retrieval protocols. It is the elapsed time between beginning of request transmission and the reception of the response message by source node. Figure 11 illustrates the average delay of proposed protocol and simple RW+PG has almost the same. The proposed protocol uses both waiting times in each hop to interrogate the neighbors and FAR_INFO wait/process which causes more queuing delay. This waiting times increases the delay even if the request has more chance than RW+PG to find the data through the farthest neighbor selection.

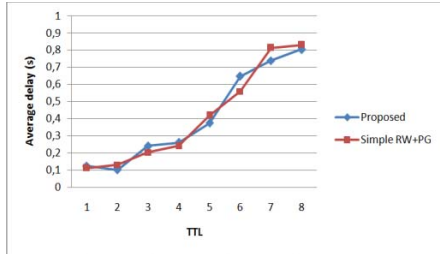


Fig. 11. Average delay of RaWPG vs. simple RW+PG.

V. CONCLUSION

In this paper, a new communicating concrete paradigm is presented for data storage and retrieval in intelligent precast. This paradigm is developed through uniformly integrating thousands of micro-sensor nodes in the concrete. The data is stored and then retrieved using multi-hop communication protocol for WSN. A previous data storage protocol called USEE is developed. However, this paper presents the data retrieval protocol. The protocol uses the multi-hop random walk process to forward the request to further nodes, and the pull gossip mechanism to interrogate the neighbor nodes in each hop. On this protocol, a mechanism called farthest neighbors selection and a cost function link were implemented. During request forwarding, only the powerful farthest neighbor is selected as next hop. This feature is very important in sensor networks since simple random walk forwards messages through randomly and low quality of service paths which causes failure in retrieval process. Moreover, the paper presents the instrumentation of the WSN inside the concrete. The adequate sensor node is chosen. Then, the density of nodes is studied for the best storage and retrieval protocols performances.

REFERENCES

- [1] R. Vrijhoef, "The four roles of supply chain management in Construction", *European Journal of Purchasing & Supply Management*, Vol. 6, No. 3, 2000, pp. 169-178.
- [2] S.H. Kosmatka, B. Kerkhoff, W.C. Panarese, "Design and Control of Concrete Mixtures", *Engineering Bulletin*, 4th edition, Portland Cement Association, USA, 2003.
- [3] P. Kalansuriya, B. Rahul, S. Sanjay, "RFID Tag Antenna-Based Sensing for Pervasive Surface Crack Detection", *IEEE Sensors Journal*, Vol. 13, No. 5, 2013, pp. 1564-1570.
- [4] LAFARGE, "Quand les ntic s'invitent sur le chantier de la tour d(2)92: Une collaboration réussie entre Lafarge et Vinci Construction France", Press release, Paris, 6 september 2013.
- [5] K. Mekki, A. Zouinhi, W. Derigent, E. Rondeau, A. Thomas, M.N. Abdelkrim, "USEE: A Uniform Data Dissemination and Energy Efficient Protocol for Communicating Materials", *Journal of Future Generation Computer Systems*, Vol. 56, March 2016, pp. 651-663.
- [6] A.P. Nikolaos, A.N. Stefanos, D.V. Dimitrios, "Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey", *IEEE Communications surveys & tutorials*, Vol. 15, No. 2, 2013, pp. 551-591.
- [7] C. Intanagonwivat, R. Govindan, D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00), August 2000, Boston, USA, pp. 56-67.

- [8] S. Li, R.K. Neeliseti, C. Liu, A. Lim, "Efficient Multi-Path protocol for Wireless Sensor Networks", *International journal of Wireless Mobile Network*, 2010, Vol. 2, pp. 110-130.
- [9] D. Ganesan, R. Govindan, S. Shenker, D. Estrin, "Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks", *Mobile Computer Communication Review*, Vol. 5, 2001, pp. 11-25.
- [10] R.C. Shah, J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks", *IEEE Conference on Wireless Communications and Networking*, March 2002, Orlando, USA.
- [11] C. Schurgers, M.B. Srivastava, "Energy efficient routing in wireless sensor networks", *IEEE Military Communications Conference*, October 2001, Washington, USA, Vol. 1, pp. 357-361.
- [12] J. Sen, A. Ukil, "An Adaptable and QoS-Aware Routing Protocol for Wireless Sensor Networks", 1st Conference on Wireless Communication, Vehicular Technology, Information Theory and Electronic Systems Technology, May 2009, Aalborg, Denmark, pp. 767-771.
- [13] F. Ye, A. Chen, S. Liu, L. Zhang, "A scalable solution to minimum cost forwarding in large sensor networks", 10th International Conference on Computer Communications and Networks, Octobre 2001, Scottsdale, AZ, USA, pp. 304-309.
- [14] F. Ye, G. Zhong, S. Lu, and L. Zhang, "Gradient broadcast: A robust data delivery protocol for large scale sensor networks", *Wireless Networks*, Vol. 11, 2005, pp. 285-298.
- [15] D. Braginsky, D. Estrin, "Rumor Routing Algorithm for Sensor Networks", the First Workshop on Sensor Networks and Applications, October 2002, Atlanta, USA.
- [16] S. Ahvar, M. Mahdavi, "EEQR: An Energy Efficient Query-Based Routing Protocol for Wireless Sensor Networks", *Journal of Advances in Computer research*, Vol. 2, No. 3, 2011, pp. 25-38.
- [17] J.B. Othman, B. Yahya, "Energy efficient and QoS based routing protocol for wireless sensor networks", *Journal of Parallel and Distributed Computing*, Vol. 70, No. 8, August 2010, pp. 849-857.
- [18] M. Maimour, "Maximally Radio-Disjoint Multipath Routing for Wireless Multimedia Sensor Networks", 4th ACM Workshop on Wireless Multimedia Networking and Performance Modeling, October 2008, Vancouver, BC, Canada.
- [19] L. Lima, L. Barros, "Random Walks on Sensor Networks", 5th IEEE Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, April 2007, Limassol, Chypre.
- [20] X. Liu, Q. Huang, Y. Zhang, "Balancing push and pull for efficient information discovery in large-scale sensor networks", *IEEE Transactions on Mobile Computing*, Vol. 6, No. 3, 2007, pp. 241-257.
- [21] J.B. Othman, B. Yahya, "Energy efficient and QoS based routing protocol for wireless sensor networks", *Journal of Parallel and Distributed Computing*, Vol. 70, No. 8, August 2010, pp. 849-857.
- [22] M. Beigl, C. Decker, A. Krohn, T. Riedel, T. Zimmer, "µParts: Low Cost Sensor Networks at Scale", 7th International Conference on Ubiquitous Computing, September 2005, Tokyo, Japan.
- [23] C. Park, J. Liu, P.H. Chou, "Eco: an Ultra-compact Low-Power Wireless Sensor Node for Real-Time Motion Monitoring", 4th Conference on Information Processing in Sensor Networks, April 2005, LA, USA.
- [24] S. Yamashita, T. Shimura, K. Aiki, K. Ara, Y. Ogata, I. Shimokawa, T. Tanaka, H. Kuriyama, K. Shimada, K. Yano, "A 15 × 15 mm, 1 µA, Reliable Sensor-Net Module: Enabling Application-Specific Nodes", 5th International Conference on Information Processing in Sensor Networks, April 2006, Nashville, TN, USA.
- [25] M. Ouwerkerk, F. Pasveer, N. Engin, "SAND: a modular application development platform for miniature wireless sensors", International Workshop on Wearable and Implantable Body Sensor Networks, April 2006, Cambridge, MA, USA.
- [26] T. Torfs, S. Sanders, C. Winters, S. Brebels, C.V. Hoof, "Wireless network of autonomous environmental sensors", 3rd IEEE Conference on Sensors, October 2004, Vienna, Austria.
- [27] S. Harte, B. O'Flynn, R.V. Martínez-Català, E. Popovici, "Wireless sensor node design for heterogeneous networks", 32nd Conference on Microelectronics and Packaging, September 2008, Pultusk, Poland.