



Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 664–671

Procedia

Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Ambient Systems, Networks and Technologies
(ANT 2017)

Towards a Three-Level Framework for IoT Redundancy Control through an Explicit Spatio-Temporal Data Model

Hedi Haddad^{a,*}, Zied Bouyahia^a, Nafaâ Jabeur^b

^a*Dhofar University, P.O. Box 2509, P.C 211, Salalah, Sultanate of Oman*

^b*German University of Technology, P.O. Box 1816, Athaibah, PC 130, Sultanate of Oman*

Abstract

In this paper we present an ongoing work towards the implementation of a framework that tackles service redundancy in IoT/WSNs as an explicit spatio-temporal phenomenon. From this perspective, redundancy is measured and explicitly stored using a spatio-temporal data model. The expected advantages of keeping an explicit history of redundancy evolution in space and time are to compare different redundancy control algorithms, to apply different knowledge extraction techniques in order to identify possible redundancy patterns, and to implement more proactive redundancy control strategies. In this paper we focus on the data model that we propose to control service redundancy at three scales: macro, meso and micro scales, respectively.

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.

Keywords: Wireless Sensor Networks and IoT; Service Redundancy Control; Explicit spatio-temporal data Modeling

1. Introduction

The challenging problem of redundancy has attracted a lot of interest in the context of Wireless Sensor Networks (WSNs), especially for its dual positive and negative impact. Indeed, on the one hand, redundancy is particularly useful for maintaining the network availability and connectivity as well as increasing network fault-tolerance and reliability. In this case, an increased redundancy makes the WSN more fault-tolerant¹. On the other hand, it leads to an accelerated depletion of the reduced sensors' energy because of duplicated data acquisition and communication

* Corresponding author. Tel.: +(968) 2323728; fax: +(968) 23237201.
E-mail address: hhaddad@du.edu.om

processes. In this case, a low rate of redundancy is commonly unfavorable as the network becomes extremely error prone. A proper redundancy rate should, therefore, be tolerated depending on the expected WSN application and its performance criteria.

Several research works have addressed the problem of node redundancy in WSNs^{2,3,4,5,6,7,8,9}. In all these approaches the concept of redundancy is commonly looked at from a low micro-level, i.e., resulting from the overlaps between sensing and/or communication ranges of neighboring sensors. In the context of Internet of Things (IoT) where sensors represent a major component, we think that this perception of redundancy is limited. We argue, indeed, that there is a necessity for a more "high-level" perception, where redundancy depends on several criteria, including the actual energy, Quality of Service (QoS) as well as current service requests. In addition, in all existing approaches the logic of how redundancy is detected and dealt with is only implicitly embedded in sensor nodes' algorithms. As such, redundancy is instantly and locally addressed by sensor nodes without retaining any explicit related information or traces of their actions. Consequently, from an outer view of the network, there is no explicit information about the extent of redundancy and how it evolved over time. We believe that having an explicit redundancy record has two main advantages. First, it provides an external reference to assess the performance of sensors and their algorithms in managing redundancy. This assessment could be benchmarked against other existing mechanisms and algorithms based on more elaborated performance indicators (such as timely-delimited redundancy measures, etc.). Second, in certain application scenarios, having an explicit redundancy record opens-up the possibility of applying data-mining techniques to detect possible redundancy patterns. In other words, redundancy could be predicted and its control can be more efficiently implemented through a proactive approach.

To address these limits, we present in this paper an ongoing work towards the implementation of a framework that tackles redundancy as an explicit spatio-temporal phenomenon. From this perspective, redundancy will be measured and explicitly stored using a spatio-temporal data model. This will allow not only to keep a history of redundancy evolution in space and time, but also to apply different knowledge extraction techniques in order to identify possible redundancy patterns, and even to use spatio-temporal visualization techniques to generate redundancy's evolution maps (redundancy cartography). To the best of our knowledge, no existing work has studied redundancy from this perspective, i.e., as an explicit spatio-temporal phenomenon, neither in the context of WSNs nor in the context of IoT. Our study is then extrapolated to the problem of service redundancy which is a more abstract concept than the problem of node redundancy. We then propose a data model and an approach to control service redundancy at three scales: macro, meso and micro scales, respectively.

In the remainder of this paper, Section 2 outlines the main concepts and functional architecture of the proposed framework. Section 3 explains the formal method that we use in order to qualitatively classify redundancy measures. Section 4 discusses our approach with regard to the related state of the art. Section 5 concludes the papers and outlines our future works.

2. Proposed Approach

In this work we address the problem of service redundancy in the context of an IoT network composed of autonomous service providers (objects) deployed in a large-scale geographic environment. As already mentioned, we propose a redundancy control framework that is based on the representation of redundancy data as an explicit spatio-temporal phenomenon. Moreover, the proposed framework addresses the problem of service redundancy at three scales, the micro, meso and macro scales, respectively. In the following we first define the main concepts of the proposed approach then we present the three-scale redundancy control scheme.

2.1. Basic concepts and notations

Modelling redundancy as an explicit spatio-temporal phenomenon requires that all the concepts of the proposed framework should be grounded to space and time dimensions. Therefore, our model is based on the concepts defined in what follows.

Spatial and temporal dimensions

We are interested to model and control service redundancy over a large spatial area of interest that we denote G . Without loss of generality, we assume a regular grid-based composition of G into n disjoint zones $z_i; i \in [1, n]$ and such that $G = \bigcup_{i=1}^n z_i$. Concerning the temporal dimension, we follow a linear time order composed of temporal points t_0, t_1, \dots, t_{end} , where $\forall 1 \leq j \leq end-1, t_{j-1} < t_j < t_{j+1}$.

Service providers

We consider a set of m autonomous service provider objects $O = \{o_1, \dots, o_m\}$, where every object $o_j \in O$ can be mobile or immobile, IP-enabled or not. Every object o_j has the capability to offer 1 or more services, from which it can decide to provide 0 or more services at any time point t . For any object $o_j \in O$, we denote by $SS_{o_j} = \{s_0, \dots, s_k\}$ the list of k services supported by o_j and by $SP_{o_j}^t \subseteq SS_{o_j}$ the set of services really provided by o_j at time t .

Service consumers

We consider a set of y autonomous service consumer objects $C = \{c_1, \dots, c_y\}$, where every object $c_j \in C$ can be mobile or immobile, IP-enabled or not. Every object c_j can consume 0 or more services at any time point t . For any object $c_j \in C$, we denote by $SC_{c_j}^t$ the set of services consumed by c_j at time t .

Spatio-temporal localization

The ability to localize the position of every object $v \in O \cup C$ at any time is a fundamental requirement to explicitly ground service redundancy in space and time. We denote by $Pos(v, t)$ the spatial position in which v is located at t . Given that not all objects are GPS-enabled, the position is defined as follows:

$$Pos(v, t) = \begin{cases} z_j \in G, & \text{if } v \text{ is not GPS-enabled} \\ (x, y) \in (\text{latitude, longitude}) \text{ or } z_j \in G, & \text{if } v \text{ is GPS-enabled} \end{cases}$$

In other words, it is possible to know the zone in which any object is located at a time t , but only GPS-enabled objects can be further located inside every zone.

Clusters

We assume that service provider objects located at a zone z_i at time t form one and only one cluster $C_{z_i}(t) = \{o_j \in O \mid Pos(o_j, t) = z_i\}$. At every time t , every cluster should have a cluster head denoted by $\bar{o}_{(z_i, t)} \in C_{z_i}(t)$. We assume that every head of cluster $\bar{o}_{(z_i, t)}$ is selected by the objects members of the cluster $C_{z_i}(t)$ based on certain mechanisms that are out the scope of this paper (see Jabeur and Haddad⁹ for further details).

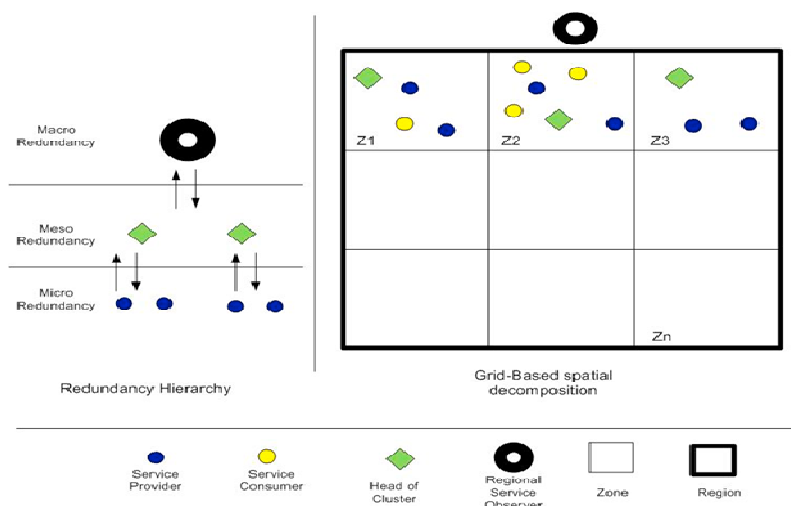


Fig. 1. A three-level redundancy control scheme

We also assume that service providers are organized according to the hierarchy illustrated in Figure 1. At the micro level, objects are autonomous, and they make their own assessment of the service redundancy based on the other service provider and / or consumer objects located in their immediate neighbouring as well on their own experiences. Based on their assessment, they decide the action(s) to be carried out in order to deal with the redundancy situation (e.g., relocate, stop offering the service, etc.). From a social perspective, service providers can be collaborative and / or competitive, depending on the application scenario. In the present work we assume a collaborative behaviour, where service provider objects are expected to inform their corresponding cluster heads about any change in their situations. At the meso level, every head of cluster of a zone z_i continuously receives and compiles data from the service providers located at that zone. It also has access to an updated information about the requests emitted by the service consumers located at that zone. From the compiled data, the head of cluster can have a knowledge about the service redundancy situation in the whole zone, which can be used to answer any request that comes from the objects in that zone or from other cluster heads. In addition, every head of cluster is requested to exchange with a Regional Service Controller (RSC), which has the responsibility of collecting and compiling data from all cluster heads in order to build a "general picture" about service redundancy at the level of the whole region of interest (macro level). The proposed three-level redundancy control scheme is presented in the following section.

2.2. Redundancy control scheme

2.2.1. Micro level

By micro-redundancy we refer to the overlaps between the communication / sensing ranges of spatially proximate objects that provide any common service s_i . We therefore adopt a physical perception of redundancy that is widely used in the state of the art. Micro-redundancy is calculated by the service providers based on their own rules. We assume that every service provider object o_j keeps a history of its micro-redundancy, in the form of data record denoted by $H_{o_j}^R$ and which is a set of temporally ordered values $v_i = \langle rl_i, \text{Pos}(o_j, t), t \rangle$, where rl_i corresponds to the numerical measure of the redundancy level during the time point t at position $\text{Pos}(o_j, t)$. As mentioned before, redundancy measures are calculated by objects, and in this work we assume that they are calculated according to the approach proposed in Jabeur et al¹⁰. In addition, and as we assume a collaborative behavior, objects are requested to update their head of clusters about any change in their situations / decisions. Particularly, they are assumed to report any change is their $H_{o_j}^R$. Also, if an object decides to relocate, to stop offering a service, or to sleep, it has to inform his head of cluster.

2.2.2. Meso level

The meso-redundancy level is calculated by the head of cluster of every zone z_i . At the meso-level we adopt a different definition of service redundancy, which is function of the ratio of free service providers on one hand and the number of waiting service requests, on the other hand. In this perception, the number of similar service providers occupying the same spatial proximity is not the real concern; the real concern is to assess this number with respect to the number of waiting service requests. For example, if the number of service providers is greater than the number of waiting service requests, we consider that there is a service redundancy. Inversely, in case the number of waiting service requests exceeds the number of service providers at a certain location, we consider that the service is not redundant. This perception can be extended later on to include more aspects of QoS, such as the service time, waiting time and service satisfaction.

Consequently, for every service s_i , at every zone z_i and at every time t , the meso-redundancy is calculated based on two data records: the number of waiting service requests and the number of free service providers.

Definition 1: The meso-History of waiting service requests of a service s_i at a zone z_i is a record denoted by $MH_{s_i, z_i}^S = \{ \langle \text{date}, s_i, z_i, t_j, nbrS_j, \phi_j \rangle \mid 0 \leq j \wedge nbrS_j \in N \wedge \phi_j \in [0, 1] \}$, where t_j is a time point, $nbrS_j$ corresponds to the number of service requests at t_j , and ϕ_j is a confidence coefficient that describes how strong do we believe that the value of $nbrS_j$ is correct.

Definition 2: The history of free service providers waiting for a service s_i at a zone z_i is a record denoted by $FO_{s_i, z_i}^S = \{ \langle \text{date}, s_i, z_i, t_j, nbrFO_j \rangle \mid 0 \leq j \wedge nbrFO_j \in N \}$, where $nbrFO_j$ represents the number of free service providers at time t_j .

Based on the two above-mentioned data records, the head of cluster of every zone z_i calculates the redundancy level at every temporal point and keeps the data in a history record as the following:

Definition 3: The history of service redundancy for a service s_i at a zone z_i is a record denoted by $R_{s_i, z_i}^S = \{ \langle date, s_i, z_i, t_j, val_j, q_j \rangle \mid 0 \leq j \wedge q_j \in R_L = \{L, M, H \mid L: \text{Low redundancy, M: Medium redundancy, H: High redundancy}\} \}$, where val_j is a numerical redundancy estimation calculated as the following:

$$val_j(t_j) = \begin{cases} \left(\frac{nbrFOj(t_j)}{nbrSj(t_j)} \right) * \varphi_j & \text{if } nbrSj(t_j) \neq 0 \\ nbrFOj(t_j) & \text{if } nbrSj(t_j) = 0 \end{cases}$$

Figure 2 illustrates an example of calculation of the meso-redundancy level of a service S_1 at a zone A. The classification of redundancy measures into qualitative classes (L, M, H) is discussed in Section 4.

MH_{s_i, z_i}^S

Date	S_i	z_i	t_j	$nbrSj$	φ_j
19-11-2016	S_1	A	10:00:00	5	1
19-11-2016	S_1	A	10:05:00	4	0.9
19-11-2016	S_1	A	10:10:00	4	1
19-11-2016	S_1	A	10:15:00	3	0.8
19-11-2016	S_1	A	10:20:00	3	1
...

FO_{s_i, z_i}

Date	S_i	z_i	t_j	$nbrFOj$
19-11-2016	S_1	A	10:00:00	3
19-11-2016	S_1	A	10:05:00	3
19-11-2016	S_1	A	10:10:00	4
19-11-2016	S_1	A	10:15:00	4
19-11-2016	S_1	A	10:20:00	4
...

R_{s_i, z_i}^S

Date	S_i	z_i	t_j	Nr_j	rl
19-11-2016	S_1	A	10:00:00	0.6	L
19-11-2016	S_1	A	10:05:00	0.675	L
19-11-2016	S_1	A	10:10:00	1	M
19-11-2016	S_1	A	10:15:00	1.06	M
19-11-2016	S_1	A	10:20:00	1.06	M
...

Fig. 2. An illustrative example of meso-redundancy calculation

2.2.3. Macro level

The macro level redundancy keeps track of the data redundancy at the level of the whole region of interest G. It is compiled and calculated, for every service s_i , by the Regional Service Controller (RSC) based on two data record types: the number of service requests of every s_i , and the history of service redundancy R_{s_i, z_i}^S generated by every head of cluster of all zones z_i . The R_{s_i, z_i}^S generated by cluster heads is already defined in Section 2.2.2.

Definition 4: The history of requests of a service $s_i \in S = \bigcup_{j=1}^m \{SSoj\}$ is denoted by $H_{s_i}^S = \{ \langle date, t_j, total_j, \{ (nb_k, z_k) : 1 \leq k \leq n \} \rangle \mid \forall j \geq 0, total_j = \sum_{k=1}^n nb_k \}$, where $total_j$ is the total number of service requests at $date$ and t_j , and the pairs (nb_k, z_k) specify how the number $total_j$ is distributed in the different zones.

Based on the above-mentioned records, the RSC calculates the redundancy at the level of the whole region of interest and generates the Regional Redundancy Record defined as follows:

Definition 5: Regional Redundancy Record is denoted by the $3R_{s_i}^S = \{ \langle day, R_L, Duration, SF, TF \rangle \}$, where $Day \in \{ \text{Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday} \}$, $R_L \in \{L, M, H\}$, SF represents the spatial features and TF the temporal features of the redundancy.

The algorithms used to generate the $3R_{s_i}^S$ are out of the scope of this paper. Figure 3 illustrates an example that shows how the $3R_{s_i}^S$ record is generated based on $H_{s_i}^S$ and R_{s_i, z_i}^S examples. In this example, the first tuple in $H_{s_i}^S$ specifies that at time 10:00:00 there is a total of 7 requests for service S_1 at 19-11-2016. These 7 requests are distributed as follows: 3 requests in zone A, 1 request in zone B and 3 requests in zone C. In the $3R_{s_i}^S$ record, redundancy data are summarized for the whole region of interest per day and for every redundancy level. In the example illustrated in Figure 3, we can see that at Saturday November the 11th, 2016, the whole region was at a low redundancy level for a total of 340 minutes. The spatial feature SP field specifies how are these 340 minutes distributed in every zone (23 minutes in zone A, 50 minutes in zone B, etc.). The temporal feature field TF details how these 340 minutes are distributed in time, so we can see that from 08:00 to 08:30 the redundancy is low in zones A and B, from 08:30 to 10:00 it is low in zones C, D and F, etc. The calculation of the spatial features and temporal features are inspired by the work of Tang et al.¹¹ about traffic modelling, and the algorithms are out of the scope of this paper.

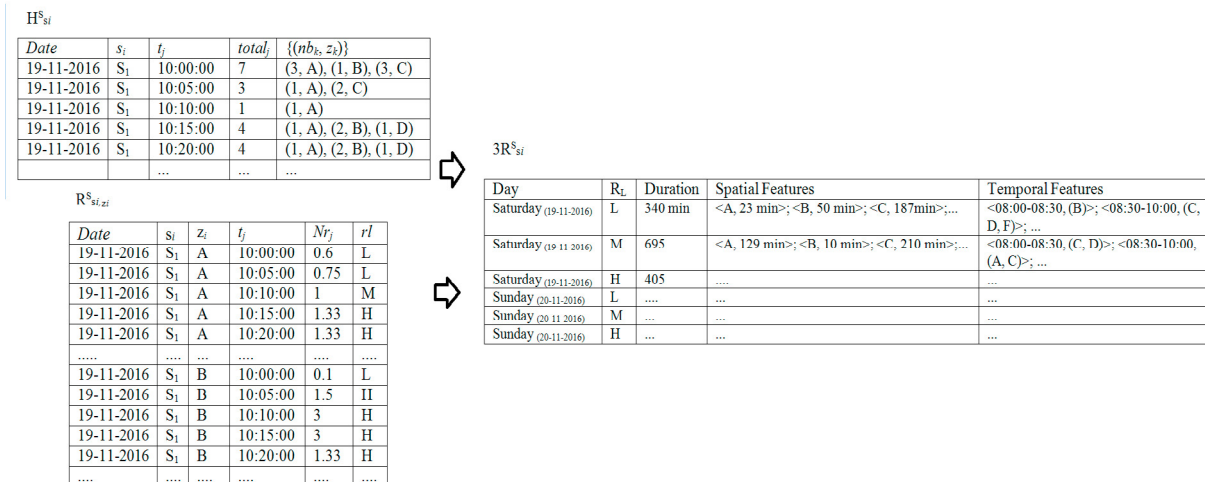


Fig. 3. An illustrative example of macro-redundancy calculation

One originality of our work is the qualification of redundancy data (Low, High and Medium levels). In fact, in the current state of the art redundancy is only measured with numerical values. By giving a qualitative interpretation to these numerical values, we intend to offer a representation that is more close to human decision makers, and to allow a more intuitive manipulation of redundancy data in the future extensions of our work. In the following section we address the problem of classifying numerical redundancy data into qualitative levels.

3. Hidden Markov Model for redundancy data classification

In this section, we outline a hidden Markov chain model for the classification of redundancy data. The main approach consists in considering the values of redundancy as an observed random process which ties together with a hidden discrete process counterpart representing a finite set of qualitative redundancy levels. The classification problem consists in matching the redundancy values with redundancy discrete levels. The number of discrete classes K corresponds to the number of levels (e.g. $K = 3$ refers to 3 levels: High (H), Medium (M) and Low (L)). Using hidden Markov chain to model redundancy allows not only an efficient joint classification scheme but also a statistical model to represent its temporal variation.

We assume that the redundancy $R_{si,zi}^N$ is a discrete time process that takes its values in \mathbb{R} . $R_{si,zi}^N = (R_{si,zi}^1, \dots, R_{si,zi}^N)$, where each $R_{si,zi}^k$, $1 \leq k \leq N$ represents the value of redundancy of service s_i in cluster z_i , N being the total number of recorded values of redundancy. We assume that each $R_{si,zi}^k$ is a real-valued random realization corresponding to a hidden process $X = (X_{si,zi}^1, \dots, X_{si,zi}^N)$. Each $X_{si,zi}^k$ belongs to a discrete set $\Omega = \{1, \dots, K\}$. For sake of simplicity we denote by r_n (resp. x_n) the value of redundancy $R_n = R_{si,zi}^n$ (resp. $X_n = X_{si,zi}^n$) at t_n . The random process X is assumed to be a hidden Markov model (HMM) relying on the following assumption:

$$p(x_{n+1}|x_n, \dots, x_1) = p(x_{n+1}|x_n)$$

The distribution of X is then defined by the distribution of X_1 denoted by $\pi_k = p(X_1 = k)$ and the transition matrices A^n , $1 \leq n \leq N$ such that $p(X_{n+1} = k|X_n = l) = a_{kl}^n$.

We also assume that the Markov model is stationary which means that transition matrix does not depend on n . The distribution of the pairwise process (R, X) can be derived in a close form from two properties:

- (i) R_n is independent conditionally to X . $p(r|x) = \prod_{n=1}^N p(r_n|x_n)$;
- (ii) $p(r_n|x) = p(r_n|x_n)$.

Whence: $p(r, x) = \pi_{x_1} p(r_1|x_1) \prod_{n=2}^N a_{x_{n-1}, x_n} p(r_n|x_n)$. The data-driven densities $p(r_n|x_n)$ are assumed to be with known mixture.

3.1. Classification algorithm

The classification process is workable in polynomial time using a MPM Bayesian criterion using the marginal *a posteriori* probabilities $\psi_n(k) = p(X_n = k|\mathbf{r})$ as follows:

$$\hat{x}_n^{MPM}(\mathbf{r}) = \arg \max_{k \in \Omega} \psi_n(k), \forall n = 1, \dots, N.$$

The computation of $\psi_n(k)$ is performed using the forward-backward procedure which consists in calculating the following probabilities: $\alpha_n(k) = p(X_n = k|r_1^n)$ and $\beta_n(k) = \frac{p(r_{n+1}^N|X_n=k)}{p(r_{n+1}^N|r_1^n)}$. These probabilities can be calculated recursively using the following computation scheme:

Boundary conditions: $\forall k \in \Omega$

$$\alpha_1(k) = \frac{\pi_k p(r_1|x_1 = k)}{\sum_{l \in \Omega} \pi_l p(r_1|x_1 = l)}$$

$$\beta_N(k) = \frac{1}{K}$$

Induction

$$\forall 2 \leq n \leq N, \forall k \in \Omega,$$

$$\alpha_n(k) = \frac{p(r_n|x_n = k) \sum_{l \in \Omega} \alpha_{n-1}(k) a_{lk}}{\sum_{i \in \Omega} p(r_n|x_n = i) \sum_{l \in \Omega} \alpha_{n-1}(l) a_{li}}$$

and

$$\forall 1 \leq n \leq N-1, \forall k \in \Omega,$$

$$\beta_n(k) = \frac{\sum_{l \in \Omega} \beta_{n+1}(k) a_{kl} p(r_{n+1}|x_{n+1} = l)}{\sum_{i \in \Omega} p(r_n|x_{n+1} = i) \sum_{l \in \Omega} \alpha_n(l) a_{li}}$$

We can easily show that $\psi_n(k) = \alpha_n(k) \beta_n(k)$.

Joint *a posteriori* probabilities $\varphi_n(k, l) = p(X_n = k, X_{n+1} = l|\mathbf{r})$ can be also calculated from forward-backward probabilities as follows:

$$\varphi_n(k, l) = \frac{\alpha_n(k) \beta_{n+1}(l) a_{kl} p(r_n|x_n = l)}{\sum_{i \in \Omega} \sum_{j \in \Omega} \alpha_n(i) \beta_{n+1}(j) a_{ij} p(r_n|x_n = j)}$$

3.2. Parameters estimation

HMM parameters can be estimated using the Expectation-Maximization procedure. The associated objective is to maximize the maximum likelihood of observed data. The EM algorithm consists in an iterative estimation of model parameters $\theta = \{\pi_k, a_{kl}, p(r|x_i = k)\}$. In each iteration, we compute forward and backward probabilities and the *a posteriori* probabilities $\varphi_n(k, l)$ and $\psi_n(k)$. HMM parameters are then estimated using the following equations:

- $\forall i \in \Omega, \hat{\pi}_i = \frac{1}{N} \sum_{n=1}^N \psi_n(i);$
- $\forall i, j \in \Omega, \hat{a}_{ij} = \frac{\sum_{n=1}^{N-1} \varphi_n(i, j)}{N \hat{\pi}_i};$
- $\forall i \in \Omega, \hat{\mu}_i = \frac{\sum_{n=1}^N \psi_n(i) r_n}{N \hat{\pi}_i};$
- $\forall i, j \in \Omega, \hat{\sigma}_{ij} = \frac{\sum_{n=1}^N \psi_n(i) (r_n - \hat{\mu}_i)^2}{N \hat{\pi}_i}, \text{ with } p(r_n|x_n = i) \sim \mathcal{N}(\mu_i, \sigma_{ij}).$

4. Discussion and related work

The problem of node redundancy has been the subject of several research works, mainly from a low micro-level perspective where redundancy is identified based on the overlaps between sensing and/or communication ranges of neighboring sensors. Many techniques have been proposed to implement this perception of redundancy. For example, Tezcan and Wang² identify redundant nodes using perimeter, center, and distance tests. Fotouhi et al.⁶

presented a computational geometry-based algorithm to estimate the area covered by the sensors in a region containing transparent and opaque obstacles and eliminate redundancy. In the algorithm proposed by Qu et al.⁸, nodes exchange their mutual information in order to classify their neighbors based on their sensing overlaps, and then apply some redundancy rules in accordance with a Boolean sensing model and a probability sensing model to detect redundancy. Le and Jang⁵ presented a scheme where nodes can estimate their coverage contribution within a given sensing area by using localization techniques. Diédié et al.⁴ presented a geometric approach to determine the area with the highest probability for a node to be redundant with respect to its neighbors' locations. Finally, Jabeur et al.¹⁰ proposed a new solution based on a comprehensive formulation of redundancy as a dynamic matter, where sensors with low redundancy weights force neighboring peers with high redundancy weights to relocate or go to sleep according to a bully strategy.

In all the above mentioned approaches, redundancy management is completely embedded in the sensor nodes, and once the network is deployed, no explicit data is kept about the network redundancy and its evolution. To the best of our knowledge, our work represents the first attempt to explicitly capture and store redundancy data, which allows for the implementation of a new approach of redundancy control in the context of IoT.

5. Conclusion

In this paper we proposed a framework for controlling service redundancy in the context of IoT. The framework is based on the perception of redundancy as an explicit spatio-temporal phenomenon, and consequently the use of spatio-temporal data model to explicitly represent and store redundancy data, with the aim of implementing a new proactive redundancy control approach. In addition, redundancy is modeled and managed at micro, meso and macro levels in accordance to three spatial scales. Constrained by the paper's length, many other fundamental components of the framework could not be presented, such as the functional architecture, the algorithms of redundancy data generation and their complexity, as well as the proactive redundancy control scheme. The performance of the framework is currently being tested with a simulation of taxi service scenario and using the Queuing theory to model certain behaviors of taxis (service providers) and customers (service consumers). Detailed results shall be presented in future extensions.

References

1. Curiac D., Volosenc C., Pescaru D., Jurca L. and Doboli A. (2009). *Redundancy and its applications in wireless sensor networks: A survey*. WSEAS Transactions on Computers, 8(4), 2009.
2. Tezcan N. and Wang W. (2007). *Effective Coverage and Connectivity Preserving in Wireless Sensor Networks*. Wireless Communications and Networking Conference, (WCNC 2007) IEEE, , 2007, pp. 3388-3393
3. Parkhi S. and Dakhore H. (2015). Redundancy Management and Energy Conservation of WSN using Multi-hop Routing and Intrusion Detection, International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 3(5). pp. 80-84
4. Diédié HG, Babri M, Oumtanaga S. (2015). Redundancy detection protocol for area coverage control in heterogeneous wireless sensor networks. *IJCSI International Journal of Computer Science Issues*, Vol.12, Issue 2, March 2015.
5. Le N-T. and Jang Y-M. (2015). Energy-Efficient Coverage Guarantees Scheduling and Routing Strategy for Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 612383, 12 pages, 2015. doi:10.1155/2015/612383.
6. Fotouhi A, Razzazi M. (2011). *Redundancy and coverage detection in wireless sensor networks in the presence of obstacles*. In MIPRO, Proceedings of the 34th International Convention, 23-27 May 2011 pp.541-546.
7. Sakib K, Tari Z, Bertok P. (2010). *An analytical framework for identifying redundant sensor nodes from a dense sensor network*. In Computer and Information Technology (ICCIT), 13th International Conference on , 23-25 Dec. 2010, pp.187-192.
8. Qu W., Wang J., and Liu Z. (2010). The Research on Redundancy Detection Algorithm for Wireless Sensor Networks. *International Journal for Information & Systems Sciences*, vol. 6, pp. 169–177, 2010.
9. Jabeur N. and Haddad, H. (2016). From Intelligent Web of Things to Social Web of Things, Facta Universitatis, Series: Electronics and Energetics, Special Issue on Internet of Things, Vol. 29(3), pp. 367-381
10. Jabeur N., Sidi Moh A-S. and Barkia M-M. (2016). A Bully Approach for Competitive Redundancy in Heterogeneous Wireless Sensor Network, The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016), pp. 628-635
11. Tang L-A., Yu X., Kim S., Han J., Peng W.C., Sun Y., Leung A. and La Porta T. (2012). Multidimensional Sensor Data Analysis in Cyber-Physical System: An Atypical Cube Approach, International Journal of Distributed Sensor Networks, Vol. 2012, article ID 724846