12th IMEKO TC1 & TC7 Joint Symposium on
Man Science & Measurement
September, 3 – 5, 2008, Annecy, France

MANAGEMENT OF COMMUNICATION IN WIRELESS INTRUMENTATION SYSTEMS: A SOLUTION BASED ON A MULTIAGENT APPROACH

Jean-paul Jamont , Michel Occello, André Lagrèze

University of Grenoble II, LCIS Labs, Valence, France, {first-name.family-name}@iut-valence.fr

Abstract: We present in this paper a multiagent approach to manage communication in wireless instrumentation system. Our solution is based on a structure emergence process. It is applied in the context of the EnvSys project which aims the instrumentation of an underground river system.

Keywords: Multiagent system, Wireless Instrumentation System, Communication Management.

1. INTODUCTION

Wireless instrumentation systems are based on wireless sensor networks. These networks are composed by autonomous hardware/software entities that achieve measuring tasks and information routing tasks. The wireless nodes have to adapt their behaviour according to their independent energy resources. In fact, in such networks, the routing process is distributed among all the nodes. Communication between two hosts is generally not direct. To communicate, entities require help from other hosts (multihop communication). Such a requirement creates an important routing problem because updating the location of neighbours is difficult. All adapted wireless routing protocols use flooding techniques. In a flooding technique, a host gives the message to all its neighbours which do the same.

Hosts have limited power resources. One of the whole system aims is so to reduce as much as possible the energy expense. When they have nothing to do generally for sparing energy they enter in a sleep mode. When they communicate they must use good routing protocols and optimal ways (generally the criteria are the number of hops). But they must decrease as much as possible the flooding scheme because the associated power cost is very high. An aggressive environment like an underground river system (as for one of our applications) can cause some internal faults for agent.

The communication infrastructure must be very adaptive, fault tolerant and self-stabilized: an agent failure must not have an important impact on the system. This system must provide reliable communications and must adapt to "real-time" constraints. Furthermore, in the case of mobile devices the infrastructure of systems are not persistent.

The ENVSYS project [1] aims the wireless instrumentation of underground river systems. Access in this type of underground galleries is difficult: it requires help from speleologists. Besides, the installation of wire communication networks is difficult, especially because the structure of hydrographical systems is very often chaotic. Finally, in the case of a radio communication network, the underground aspect complicates wave propagation and, at this day, the techniques which are used are not totally mastered. In this type of network, a sensor interacts with the environment to carry out a measure and help other sensors to communicate with the master station which collects all sensor data from inside the cave. The main contribution of the work presented in this paper is situated at a logical level, concerning the energetic safety routing management...

A multiagent approach. Multiagent systems are especially adapted for designing complex systems. Through the MWAC (Multi-Wireless-Agent-Communication) model, we propose an innovative approach for open multiagent systems in the context of wireless networks of intelligent sensors. The cooperative, collaborative and negotiation capabilities of MAS allow the agents which evolve in an open system to increase the overall efficiency of the whole system.

Organization of the paper We introduce in a first part the multiagent paradigm and the emergence feature. Then, we give an insight to our practical problem (the EnvSys project), to the main difficulties for this kind of application and to the solutions traditionally used to solve this problem. The EnvSys Project (This project is funded by the FITT program (Fund for Technological Transfer) of the French Rhône-Alpes Regional Council) aims the instrumentation of an underground river system.

We then present our model. We show how we improve both management of communication and management of energy resources including a fault tolerant feature. We explain how of a totally decentralized approach and the inherent multiagent emergence features are exploited. Before to conclude, we give some evaluation results validating our approach.

2. EMERGENCE AND MULTIAGENT SYSTEMS

Multiagent systems. An agent is a software entity embedded in an environment in which it can perceive and in which it acts. It is endowed with autonomous behaviours and has objectives. Autonomy is the main concept in the agent issue: it is the ability of agents to control their actions and their internal states. The autonomy of agents implies no centralized control.

A multiagent system is a set of agents situated in a common environment, which interacts and attempts to reach a set of goals. Through these interactions a global behaviour, more intelligent than the sum of the local intelligence of the agents, can emerge. The emergence process is a way to obtain, through cooperation, dynamic results that cannot be calculated in a deterministic way.

Emergence. The emergence paradigm deals with the not programmed and irreversible sudden appearance of phenomena in a system confirming that "the whole is more than the sum of each part". It is one of the expressions of collective intelligence [2]. The emergence process is a way to obtain from cooperation dynamic results that cannot be predicted in a deterministic way. There are three types of emerging features [3]: emergence of structures at the origin of the self-organization process, emergence of behavior and emergence of properties. It is difficult to qualify the emergent characteristics of a phenomenon. Some fundamental elements have been settled by S. Forrest [4]. J.P. Muller [5] proposes an interesting specialization in the multi-agent context that has been recently discussed and completed by Dessales and Phan [6]. They affirm that a phenomenon is emergent if:

- there is a set of agents interacting via an environment whose state and dynamic cannot be expressed in terms of the emerging phenomenon to produce in a vocabulary or a theory D,
- the dynamic of the interacting agents produces a global phenomenon such as, for example, an execution trace or an invariant,
- the global phenomenon is observable either by the agent (strong sense) or by an external observer (weak sense) in different terms from the subjacent dynamics i.e. another vocabulary or another theory D '.

To give to a system of agents a particular global functionality, the traditional method consists in carrying out a functional decomposition of the problem into a set of primitives which will be embodied in the agents. The alternative suggested by L. Steels [7] aims to make this functionality emerge from the interactions between the agents.

The advantages of the "emergent functionality" approach are first of all a reinforcement of the robustness of the system: it is less sensitive to the changes of the environment. The reason is that, unlike to the case of a programmed functionality (traditional approach), the designer doesn't need to consider all the possibilities for the system react according to each situation.

4. OUR SOLUTION MULTIAGENT APPROACH

Our objective is to decrease the energy expense induced by the inherent flooding techniques. We use a MAS approach to implement an emergence of structure. In this section, we will discuss the basic structures that we want to see emerge.

4.1. What should emerge?

Our organizational basic structures are constituted by (see fig. 1): one and only one *group representative agent* (r) managing the communication in its group, some *connection agents* (c) which know the different representative agents and can belong to several groups and some *simple members* (s) which are active in the communication process only for their own tasks (They don't provide information relay).

With this type of organizational structure, the message path between the source (a) and the receiver (b) is ((a,r) , *[(r,c),(c,r)] , (r,b)). If the source is a representative agent the first term doesn't exist. If the receiver is a representative agent the last term doesn't exist.

The energy saving comes owing to the fact that the flooding is only directed to the representative agent of the groups and to some connection agent. To give an order of idea, a receiver path research with flooding techniques will cost, in the case of a traditional wireless network, a number of emissions equal to the number of stations. In the case of a clustered wireless network, the number of transmitted

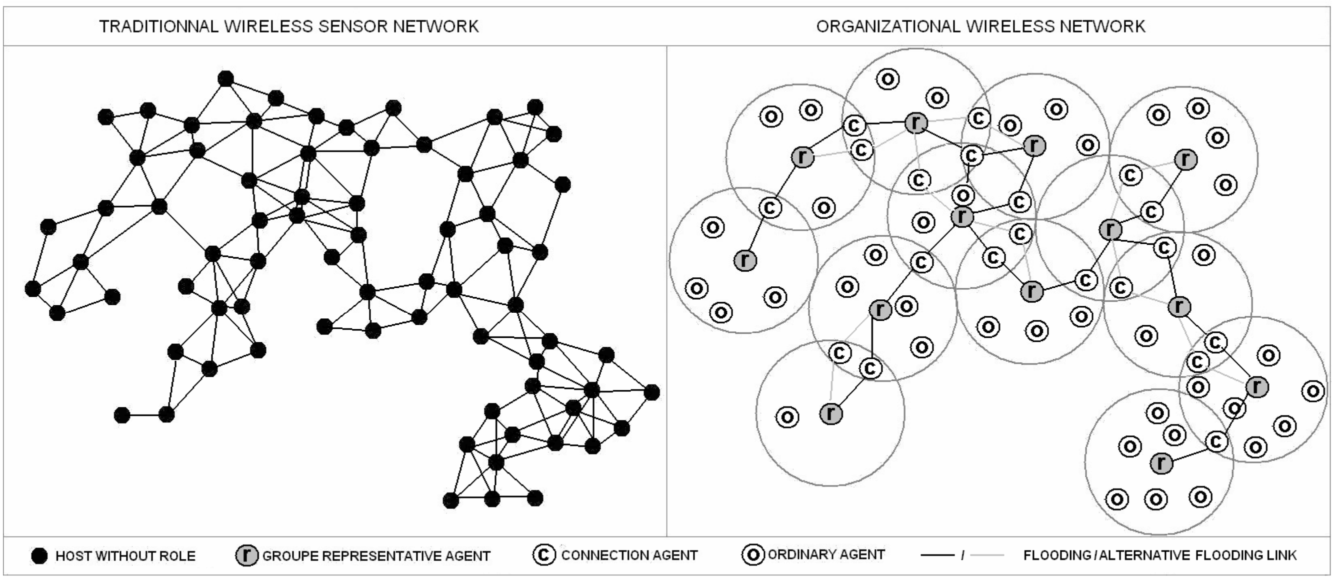

Fig. 1. Our organizational structure

messages is about twice the numbers of representative agent (all the representative agents are contacted via one connection agent).

However, the networks with an organizational structure must take care of the maintenance of their routing table. Generally, the adaptive features of these tables come from periodical exchanges between the different nodes. In our approach we do not wish to use this technique to ensure the maintenance of coherence. Indeed, our principle will be "if we do not need to communicate, it is useless to spend energy to ensure the coherence maintenance". However, we will thus use eavesdropping of surrounding agent communications. We extract from these messages exchange knowledge to update our beliefs about our neighbours. Moreover, our self-organization mechanism will integrate an energy management policy. These structures will thus emerge.

4.2. How to make the solution emerge?

It is necessary for us to wonder now how we will make emerge these structures. The multiagent methods aim at decreasing the complexity of system design by a decentralized analysis. We focus here on the *organization view* of MAS. This view allows ordering agent groups in organization determined according to their roles.

We want obtain an adaptation of our whole MAS through the emergence of organizational structures by self-organization based on role allocation modifications. The organization is built according to an exchange of messages between agents. Relations between agents are going to emerge from the evolution of the agents' states and from their interactions. We are only going fix the organization parameters, i.e. tasks of agents, roles of agents.

The ideal representative agent is the one having the most important number of neighbours and the most important level of energy. The level of energy is an important parameter in the sense that the representative agent is the most solicited agent in the group from a communication point of view. We use role allocation based self-organization mechanisms involving the representative agent election. Our election function integrates some data on neighbours and on energy level. This function estimates the correlation between its desire to be the boss and its capacity to access to this position. The organization is modified only when a problem occurs. We do not try to maintain an organization if there is not communication. In addition to the configuration messages, all agents use eavesdropping. In fact, when some communicating entities (humans, robots etc.) share a common environment they might intercept some messages (broadcasted or not). From this eavesdropping message they can extract some authorised information like the receiver, the sender, the type of message and the packet's path.

We propose here a formalized description of our model. The notation finds their sources in the work described in [7].

Identifier. Hosts of the network are modelled by agents. We note A_i the agent identified by i.

The multiagent system. The multiagent system Γ is the set of agents $\Gamma = \{A_1, A_2, ..., A_i, ..., A_n\}$ with $card(\Gamma) = n$. Our MAS is open: hosts can enter or leave the system.

Time. We note $\mathbb{T}$ the ordered set with the operator > and an element $-\infty$ with $\forall t \in \mathbb{T}, t > -\infty$. So $\mathbb{T} = \mathbb{N} \cup \{-\infty\}$.

Groups. *(1)* An agent group is noted G. In our organization, a group is identified by its representative agent Identifier. The group where the representative agent is A_R is noted G_R. All groups are part of the system: $G_R \in P(\Gamma)$.
(2: intention) A group has a finite time to live (with a lower and a higher limit). The lower limit is the most interesting (the group birth): we note $[A_R, t_0]$ the group created by A_R at t_0 with $(A_R, t_0) \in \Gamma \times \mathbb{T}$.
(3: belief in extension) We note $[A_R, t_0]^{A_j, t_1}$ the set of agents that A_j think members of the group $[A_R, t_0]$ at t_1.
(4: extension) We note $[A_R, t_0]^t$ the set of agents really in $[A_R, t_0]$ at t. We note $[A_R, t_0]^t$ the group composition G_R created at t_0 at the given date t. This knowledge can be defined from the belief of the agents:

$$[A_R, t_0]^t = \{A_j \in \Gamma / A_j \in [A_R, t_0]^{A_j, t} \wedge A_j \in [A_R, t_0]^{A_R, t}\}$$

Belief. $\mathcal{B}_{A_i} \varphi$ believes that the agent A_i thinks φ, in other words it thinks that φ is true. To highlight the recursive feature of the group definition given previously, we can note that :

$$(A_j \in [A_R, t_0]^{A_i, t}) \equiv (\mathcal{B}_{A_i}(A_j \in [A_R, t_0]^t))$$

Desire. $\mathcal{D}_{A_i} \varphi$ minds that the agent A_i desires φ, in other words it wants to verify φ.

Knowledge. $\mathcal{K}_{A_i} \varphi$ minds that the agent A_i knows φ.

Roles. *(1)* We note $role(A_i, t)$ the function that returns the role of the agent A_i at the date t with $(A_i, t) \in \Gamma \times \mathbb{T}$. A role can be R_R for a representative agent, R_C for a connection agent and R_S for a simple member. When an agent is initialized, he has no role. The function $role$ can then return $\emptyset$ to signify that the agent has no role.
(2: simplification of writing) We note $role_t(A_i)$ the last role taken by A_i.
(3: choice of a role) Each agent chooses a role depending on its neighbourhood. So, choosing a role leads to notify the new role to neighbours and modify its knowledge about its own role. So $\mathcal{K}_{A_i}(role(A_i, t_v) = R_R)$ can be understood following different ways.
Firstly, we learn simply that A_i is a representative agent.
Secondly, if $\mathcal{K}_{A_i}(role(A_i, t_{v-1}) \neq role(A_i, t_v))$ then the agent A_i has modified his role to be representative.

Groups. Similarly to the function $role$ it exists the function $group$ which returns the group identifier of an agent.

Power supply. *(1)* We note $power(A_i, t)$ the function which returns the energy level (a percentage) of the agent A_i at the date t with $(A_i, t) \in \Gamma \times \mathbb{T}$.
(2: simplification of writing) We note $power_t(A_i)$ the current energy level of the agent A_i.

Neighbourhood. We note N_{A_i} the neighbourhood that A_i knows. It is a set of agents in the emission range of the agent A_i not including itself $(N_{A_i} \in P(\Gamma))$. An agent knows a neighbour by its unique identifier but it can access to its role and its group: $(\forall A_j \in N_{A_i},\ \mathcal{K}_{A_i}\, role(A_j) \wedge \mathcal{K}_{A_i}\, group(A_j))$. We can notice that if $\mathcal{K}_{A_i}[A_R, t_0]^{A_j, t_1}$ then $\mathcal{K}_{A_i}\, group(A_j) = R$. The reciprocal is not true because there is an uncertainty about the time.

Formalized description of the role attribution. Choosing a role depends firstly on its neighbourhood (*basic algorithm*). However, because its power level is low, an agent can not desire to be a representative (*energetic constraint*).
The decision processes of agents are not synchronized. Two neighbours can take the same decision at the same time.
It is possible that two close agents choose a representative role: there is a *representative conflict* which must be detected and corrected. It is possible to have two closer groups which don't include a connection agent between them: there is an *inconsistency* which must be detected and corrected.
We begin by focusing on our algorithm which allows to the agent A_i to choose a role in function of its neighbourhood N_{A_i}

Basic algorithm. *1)* There is no neighbour: the concept of role doesn't make sense:

$$(N_{A_i} = \phi) \Rightarrow (\mathcal{K}_{A_i}(role(A_i) = \phi))$$

2) Neighbours exist $(N_{A_i} \neq \phi)$.

$$\mathcal{K}_{A_i}(card(\{A_j \in N_{A_i} / role(A_j) = R_R\}) = 0) \Rightarrow (\mathcal{K}_{A_i}(role(A_i) = R_R))$$

$$\mathcal{K}_{A_i}(card(\{A_j \in N_{A_i} / role(A_j) = R_R\}) = 1) \Rightarrow (\mathcal{K}_{A_i}(role(A_i) = R_S))$$

$$\mathcal{K}_{A_i}(card(\{A_j \in N_{A_i} / role(A_j) = R_R\}) > 1) \Rightarrow (\mathcal{K}_{A_i}(role(A_i) = R_C))$$

Energetic constraint. Generally, the role of representative or connection makes that the agents take an active part in the management of communications. From this fact, consumption of energy is higher. So,

$$(power(A_i) < trigValue) \Rightarrow (\mathcal{K}_{A_i}(role(A_i) = R_S))$$

Detecting and correcting a representative conflict.
(1: Conflict detection) An agent A_i detects a conflict with others agents if $\mathcal{K}_{A_i}(N_{A_i} \neq \phi) \wedge \mathcal{K}_{A_i}(role(A_i) = R_R) \wedge \mathcal{K}_{A_i}(card(\{A_j \in N_{A_i} / role(A_j) = R_R\}) \geq 1)$
(2: Conflict correction) A_i has detected a conflict with other agents. It sends a `ConflictRepresentativeResolution` message (see the interaction aspect) to its representative neighbours. This message contains the score of the agent A_i. The agents, which receive this message, calculate their own score. Agents with an inferior score leave their role and choose another. An agent with a better score sends its score to its neighbours.
An example of `score` function can be simply expressed. The following function supports an agent with a high energy level and a significant neighbour (the interest is to have dense groups in order to limit the flooding volume).

$$score(A_i) = power(A_i).card(N_{A_i})$$

Detecting and correcting an inconsistency
(1: Inconsistency detection) An inconsistency in the organization can be detected only by one representative starting from beliefs of one of its members. This detection needs an interaction between an agent A_i and its representative A_R (`VerifyNeighbourGroupConsistency` message).
The agent A_i will send the list of the groups of its neighbourhood of which it does not know if its representative knows the proximity.
We define $N_{A_i,C} = \{A_k \in N_{A_i} / role(A_k) = R_c\}$.
A connection agent is member of many groups, so, if $A_C \in N_{A_i} \wedge role(A_C) = R_c \wedge [A_\alpha, t_\alpha]^{A_C, t_\alpha} \wedge [A_\beta, t_\beta]^{A_C, t_\beta}$ then $\mathcal{K}_{A_i}(group(A_C) = \alpha)$ and $\mathcal{K}_{A_i}(group(A_C) = \beta)$
We define:

$$\zeta_{A_i} = \{ A_j \in N_{A_i} / (group(A_j) \neq group(A_i)) \wedge ((\neg\exists A_k \in N_{A_i}, L / (group(A_k) = group(A_i)) \wedge group(A_k) = group(A_j)) \}$$

The inconsistency is found by A_i if $card(\zeta_{A_i}) = 0$. The representative agent A_R of A_i receives a message with ζ_{A_i}. For $\forall A_n \in \zeta_{A_i}$, if $card(\{A_y \in N_{A_R,C} / group(A_y) = n\}) = 0$ then there is a real inconsistency.
(2: Inconsistency correction) In this case several strategies can be used. We judge that if a path with a low energy cost is available, one will support a stability of the organization to reorganization. A search for path towards one of the groups soft will thus be sent with a TTL (Time to Live) relatively low.
If a path exists, the organization does not change. If not, the representative proposes to A_i, if $role(A_i) = R_c$, to be a representative (`ISuggestYouToBeRepresentative` message). The agent A_i can refuse to become representative (if its energy level is too low) but in all the case, the representative A_R leaves its role.

4. USING THIS MULTIAGENT SOLUTION

We have built the MWAC middleware (Multi-Wireless-Agent Communication fig. 2) based on the self-organized multiagent solution expressed above. This mobile communication management layer to manage the wireless communications between the different agents of the system. This layer must increase interoperability, portability and flexibility of an application by allowing the application to be distributed over heterogeneous multiple agents. It must reduce the complexity of development of the agents. This layer is a Message Oriented Middleware (MOM).

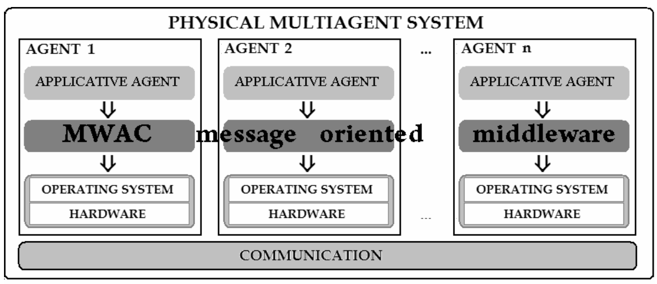

Fig. 2. The MWAC middleware

The MAS middleware services are supplied through a software component. The agent must use a `WCommunication` package, written in Java language and translated into C++ language because a lot of physical platforms frameworks use this language. This package contents two abstracts classes (`Identifier` and `Message`) and two main classes called `Communication` and `BitField`.

In the `Message`} abstract class the designer must implement the primitives to convert the message in a bit field `BitField MessageToBitField(Message m)` and the reciprocal primitive `Message BitFieldToMessage(BitField b)`). In the Identifier abstract class the designer must implement the type of identifier and two primitives `BitField IdentifierToBitField()` and `Message BitFieldToMessage(BitField b)`. The primitive to convert the identifier in a bit field must be implemented by the designer. The Communication class contains a list of couples `(Identifier, Message)` for the emission and the reception. This list is private and must be accessed via `SendMessage(identifier, Message)` and `CoupleIdentifierMessage ReceiveMessage()`.

The package must be connected to the operating system. The operating system must give the battery energy level (primitive `SetBatteryLevel(Float l)` to the `Communication` class and must give the bit field which arrives. In another hand, the middleware gives to the operating system the bit field to send by calling `BitField GetBitFieldToSend()`.

These agents are embedded on autonomous processor cards. These cards are equipped with communication modules and with measuring modules to carry out agent tasks relative to the instrumentation. These cards supply a real time kernel. The KR-51(the kernel's name) allows multi-task software engineering for C515C microcontroller. We can produce one task for one capability. We can then quite easily implement the parallelism inherent to agents and satisfy the real-time constraints.

In the EnvSys project, we use the physical layer which is employed by NICOLA system, a voice transmission system used by the French speleological rescue teams [8]. This layer is implemented in a digital signal processor rather than a full analogical system. Thereby we can keep a good flexibility and further we will be able to apply a signal processing algorithm to improve the data transmission.

The link layer used is a wireless version of the CAN (Controller Area Network) protocol stemming from the motorcar industry and chosen for its good reliability.

The applicative layer is constituted by the agents of the system. A hybrid architecture has been chose. It enables to combine the strong features of each of reactive (to the message) and cognitive capabilities (to detect inconsistency and re-organisation). The ASTRO hybrid architecture [9] is especially adapted to a real time context. The integration of deliberative and reactive capabilities is possible through the use of parallelism in the structure of the agent. Separating Reasoning/Adaptation and Perception/Communication tasks allows a continuous supervision of the evolution of the environment. The reasoning model of this agent is based on the Perception/Decision/Reasoning/Action paradigm. The cognitive reasoning is thus preserved, and predicted events contribute to the normal progress of the reasoning process.

5. EVALUATION OF OUR APPROACH

In order to evaluate and improve such agents' software architectures and the cooperation techniques that they involve, we introduce a simulation stage in our development process. The simulation first allowed us to experiment our approach and the software solutions that we provide for the various problems.

The simulation first allowed us to experiment our approach and the software solutions that we provide for the various problems. We can also quantify the emergence inferred by the MAS approach in this case. Our platform provides some functionalities:

- to create the environment i.e. to choose a map and to determine the geographical position of the sensors,
- to spy a sensor i.e. to have access to its attributes (energy level, role, range) and have its various tables of routings,
- to measure and visualize the performances of the system to various criteria like the efficiency, the average used memory etc,
- to create faults like to erase the agent's data or to remove it in order to study the fault tolerance of our system,
- to create displacements in order to study the impact of mobility on the efficiency of the studied techniques,
- to allow the creation of scenario because for each tested protocol we must be sure that events occur at the same time.

We have compared our MAS to three traditional solution based on ad-hoc protocols. The DSDV protocol (Destination-Sequenced Distance-Vector protocol [10]) and the natural DSR protocol (Dynamic Source Routing protocol [11]) do not appear in this comparison because its efficiency were lower than the enhanced version of DSR which uses a route maintenance (memorization of the main route).

We thereafter call efficiency the ratio between the theoretical useful volume of the optimal way divided by the volume of each transmitted communication. The EnvSys project evolves unidirectional communication: all the sensors communicate only with the workstation situated at the exit of the underground river system. In this case, messages are small (it is an advantage for our solution because we use non optimal path in hop term).

In our simulation three messages are sent by five seconds. The same scenario is applied for the different protocols. We can see that the benefit (fig3) of our approach is important in the EnvSys case. We have drawn the ratio between the volumes transmitted in the whole system with DSR by the whole volume transmitted with MWAC.

Our routing method can deliver quickly all messages with a good efficiency. Higher is the number of sensors better is the reactivity of our approach. We must note that if the system knows no perturbation or mobility variation of DSR will be better from an efficiency point of view is normal because in this case DSR learns all the routes (succession of sensors) allowing to communicate with the workstation. It is not really the case of our approach: a route is a succession of group. One consequence is that the routes used by the messages with our approach are not optimal.

We can see that our approach supports the addition of a lot of sensors. The number of groups doesn't explode with the number of sensors but their density increases.

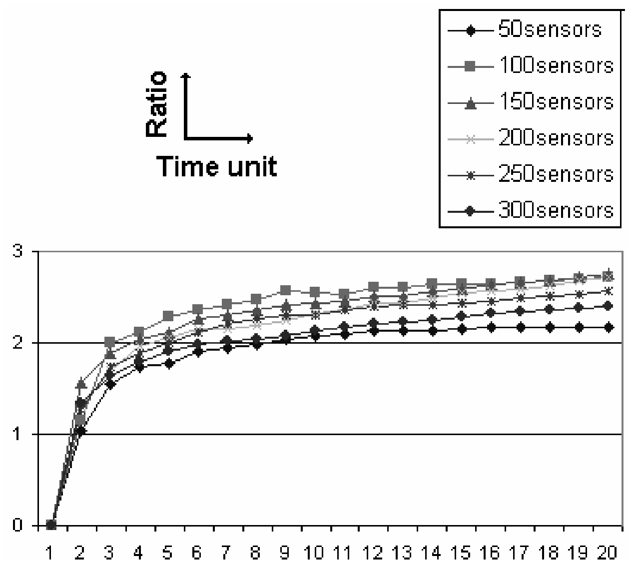

Fig. 3. DSR/MWAC transmitted volume

6. CONCLUSION

In this article, we have presented a multiagent system to manage communication in wireless instrumentation systems in respect to energetic constraints. We use structure emergence and collective features to make the system adaptive. This paper wants to contribute to show that AI mechanisms as organizational structure emergence can lead to interesting results and can improve classical techniques.

This middleware allows managing the openness of the system: adding an host does not require a manual reconfiguration. Most of hosts' dysfunctions should not threaten the functional integrity of the whole system: it is be self-adaptive to a sensor power fault.

This model is used in the EnvSys project [1] and in an effective flood decision support system that can help scientific to limit flood damage [12].

REFERENCES

[1] J.-P. Jamont, M. Occello, André Lagreze, A Multi-agent System for the instrumentation of an underground hydrographic system, IEEE International Symposium on Virtual and Intelligent Measurement Systems, IEEE Instrumentation and Measurement Society, 2002.

[2] J. Deguet, Y. Demazeau, and L. Magnin, Elements about the emergence issue: A survey of emergence definitions," Complexus, vol. 3, no. 1-3, pp. 24-31, 2006.

[3] P. Marcenac, Emergence of behaviors in natural phenomena agent-simulation," in Complexity International Review, vol. 3, 1996.

[4] S. Forrest, Emergent computation. The MIT Press, 1991.

[5] J.P. Muller. Emergence of collective behaviour and problem solving," in Engineering Societies in the Agents World, LNCS 3071, pp. 1-21, Springer, 2004.

[6] J.L. Dessalles and D. Phan, Emergence in multi-agent systems: cognitive hierarchy, detection, and complexity reduction," Society for Computational Economics, Computing in Economics and Finance, Nov. 2005.

[7] F. Legras, C. Tessier: LOTTO: group formation by overhearing in large teams, International Joint Conference on Autonomous Agents & Multiagent Systems: 425-432, 2002.

[8] G. Naylor, Add-ons for the Nicola System, in The Cave Radio and Electronics Group Journal, vol. 44, pp. 11-12, December 2001.

[9] M. Occello, Y. Demazeau, and C. Baeijs, Designing Organized Agents for Cooperation in a Real Time Context, Proceedings of the first International Workshop on Collective Robotics, LNCS/LNAI 1456, Springer-Verlag, p25-73, 1998.

[10] C. E. Perkins and Royer, E.M. and S. Das, Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing for Mobile Computers, ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications, ACM, 1994.

[11] D.-B. Johnson and D.-A. Maltz, Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing, Kluwer Academic Publishers, pp. 153-181, 1996.

[12] J. Luo, L. Xu, J.-P. Jamont, L. Zeng, Z. Shi, Flood decision support system on agent grid: method and implementation, Enterprise Information Systems, vol. 1, n°1, Taylor & Francis, 2007.