

Dynamic Reconfiguration of LPWANs Pervasive System using Multi-agent Approach

Ghouthi ABDELLAOUI, Fethi Tarik BENDIMERAD
University Abou Beker Belkaid,
Faculty of Technology
Tlemcen, Algeria

Abstract—The development of the Low Power Wide Area Network (LPWAN) has given new hope for the Internet of Things and M2M networks to become the most prevalent network type in industrial world in the near future. This type of network is designed to connect several entities in a radius that can reach up to 10 Km. This gain in scope is possible through a reduction in the amount of information exchanged. The latter will makes LPWANs Networks the most suitable for telemetry applications. This large network coverage offered by LPWAN gives the possibility to connect a large number of objects. On the other hand, it involves difficulties for the pervasive system associated with this kind of network to integrate dynamically all this objects, in which an automatic reconfiguration process becomes crucial to this kind of network. In this study, we propose the multi-agent systems as a solution to virtualize the heterogeneity of the peripherals and to facilitate their integration and their dynamic exploitation in the LPWAN system. That virtualization is possible due to the portability of the multi-agent systems and the standardization of the exploitation of the services offered by them.

Keywords—Dynamic reconfiguration; pervasive system; multi-agent system; Low Power Wide Area Network (LPWAN)

I. INTRODUCTION

The technology revolution has created a new world dedicated to information processing, where any real component is characterized by two sets (data and Handling). This can be seen as a global observing environment for several systems in different areas such as control of medical equipment, autonomous support, control and traffic safety, advanced automotive systems, process control, energy conservation, environmental control, control of critical infrastructure (electricity, communications system, e.g. tele-presence telemedicine), military systems, manufacturing systems, and any smart structures, etc. The majority of these systems are critical and vary chaotically in time, which require continuous changes of their representation in the information world.

The Low Power Wide Area Network (LPWAN) [1] such as LoRaWan [2] or SigFox [3] provides a solution to connect each component or object to the network and monitor their behaviour in real time through their pervasive system layer. For this purpose, we need to develop a mechanism for observing and dynamically updating the system, which can be considered as a dynamic reconfiguration for the initial state of the LPWAN pervasive system. This mechanism could be associated to the pervasive system layer of the LPWAN network, which is responsible for identifying and recognizing

all the objects present in the network.

In this paper, to give pervasive system better perception of environment and make him more interacting “naturally” with the user, we propose a new mechanism improving the context-aware of this latter. For that, the systems must adapt to the different changes (adding and/or deleting a new subsystem) and dynamically reconfigure the changes that are imposed by the environment.

We will use the multi-agent systems approach [4], [5] to perform the dynamic reconfiguration. This chose is made, because they present better portability and flexibility by dint of the different behaviours offered by them, which gives a great capacity to improve context-aware of a pervasive system.

The rest of the paper is organized as follows. In the next one (Section II), we present the state of the art of the study. In this section, the dynamic reconfiguration and some reasons to do it with their approaches are given, in the second one, we presents the pervasive system and the most its important characteristics. After that, we talks about multi-agent system and what is the aim to use theme for the dynamic reconfiguration. This section will be end by the presentation of the multi-agent system as solution for our problematic. Section III gives the LPWAN networks like a case of use to implement our proposed approach in order to perform the dynamic reconfiguration of the pervasive system. Section IV presents our implementation for the proposed approach. Section V discuss the use of this approach and their advantages. Finally, Section VI draw the context of our work and future scope.

II. STATE OF THE ART

Along this section we give an overview of the dynamic reconfiguration and their different approaches and the multi-agent system as a solution to perform a dynamic reconfiguration on the LPWAN networks.

A. Dynamic Reconfiguration

Currently, many applications require a runtime dynamic reconfiguration. This requirement becomes critical when it is not possible to stop or restart a component or a service.

There are several examples in which the application of this practice is crucial. Especially in the industrial area where the embedded systems is complex, such as operating system of some modern cars control units requires operating systems while different modes with several possible configurations. In this case, we can said that the reconfiguration is the set of passage operations from one configuration to another, and we can take the following definition of the dynamic reconfiguration: "Dynamic reconfiguration is the ability of a software system to allow modification of a subset of the system during its execution, without interruption of service" [6].

In some areas, it is enough to have a finite set of configurations like the case of cars, but in others ones, we cannot anticipate all possible configurations. For this reason, the use of an approach based on configuration rules is more appropriate.

Among the applications which have trend towards dynamic reconfiguration, applications based on the agent approach are involved. This one becomes the most used method in the world of software development due its benefits to the use.

1) Reasons for the reconfiguration: Each system has its own reasons to be reconfigured according to the abnormalities, this may cause undesired operation. Some of ones, such as distributed system, risk of failures at server or become obsolete. Such a system requires a reconfiguration to correct this problem as in the case of zookeeper system [7]. In order to perform the composition of several web services and extend it more and more. This latte need to be changed as well performed and adapted to new ISC [8] specifications. In this approach the main element of the system is the service, since each one has its own specifications and works independently from the others.

The dynamic reconfiguration can be used to expand embedded systems with the reuse of hardware resources through a middle ware [9], or by the use of systems based on FPGAs [10], whose main characteristic is the dynamic reconfiguration.

The reconfiguration of FPGAs is possible through the ability to reprogram some blocks (CLB) [10] that are elected uncomfortable for the application. Knowing that each block or group of blocks can implement one or more spots in the application, this makes it modular and therefore the runtime partial maintenance is possible.

Also a dynamic reconfiguration can be done for several reasons (correction, adaptation, improvement); this involves the integration or deletion of system data, giving a new state for the system; this transition from one state to another gives a dynamic aspect to the system and therefore good reflection for the real system. The reconfiguration of one system will be the transition between two states.

2) The different approaches for the reconfiguration process: There are many approaches for the reconfiguration of systems due to their constraints and their constituents elements. The following paragraphs present some approaches for the reconfiguration:

a) Reconfiguration for the primary/backup system [7]: The reconfiguration in this approach is the transition between a source (S) and a target (S') configuration, through the following steps:

- Persist information about S' on stable storage at a quorum of S.
- Deactivate S.
- Identify and transfer all committed S to S', persisting it on stable storage.
- Activate S'.

b) The reconfiguration steps to reuse embedded system resources [9]: The following steps show how to reuse embedded system resources or add embedded hardware device or add function to the middle ware:

- Analysing the embedded system resources.
- Dynamically loading them to the middle ware.
- Registering them on middle ware.

c) The reconfiguration steps for the component software approach [6]: Intuitively reconfiguration takes place as follows:

- Identify and locate the part of the system to be reconfigured.
- Suspend execution (to avoid corrupting the system).
- Change the configuration of the system (add, remove parts ...).
- Transfer the report to the new parts.
- Resume execution of the interrupted part of the system.

These three systems seem to be different, however their abstraction notice that they target the same thing (reconfigure), even with their different units of composition, but they share the same properties:

- Elemental.
- Independent.
- Connectible.
- Maintainable.

For this aim, our approach will be based on these properties to meet a large number of systems.

There are several technologies which can ensure these properties such as component programming (OSGIs) [11], web services [12], software agents, etc. all these technologies which are based on independent elementary entities, connectable and easily maintainable, except the software agents which has another feature that makes them closer to reality and allows us

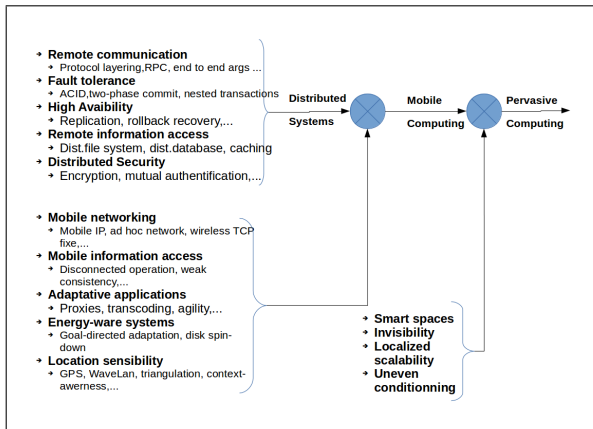


Fig. 1. Taxonomy of computer systems research problems in pervasive computing [14].

to create a copy of the real world in the virtual one (digital). This characteristic is offered by the behaviours of agent, which can effectively ensure the two main characteristics of a pervasive system (context-aware and ambient). Moreover, the software engineering for multi-agent systems can provide powerful techniques, methods and tools for the engineering of modelling and developing pervasive systems.

In the following sections, we present how we can use the software agents to perform dynamic reconfiguration of pervasive systems.

B. Pervasive System

In 1991, Mark Weiser [13] propose a new vision for computer systems; it was represented in the form of ubiquitous systems which will be developed later in pervasive systems. And his vision was defined as follow: “A new way of thinking about computers in the world, one that takes into account the natural human environment and allows the computers themselves to vanish in the background” [13].

In other words he wanted to create an environment with a computing capacity and communication. This vision has gone through various steps in order to arrive at the pervasive systems, which can be summarized as follows:

- Ubiquitously: Accessible anywhere over the network.
- Mobile: Integrate mobile terminals.
- Context-aware: Take the execution context into account.
- Ambient: Embedded in everyday objects.

So, pervasive systems [14] are therefore ubiquitous, mobile and context-aware systems as shown in Fig. 1. And the evolution of the pervasive systems depends directly on the evolution of the latter.

C. Multi-agents System

Multi-agent systems (MAS) provide abstractions that allow decomposing a system to a set of agents including distributed

systems; Multi-agent systems provide flexibility for modelling more sophisticated, globally emergent behaviour; Multi-agent systems deploy services throughout the network; Multi-agent systems by their nature are powerful tools for modelling complex systems.

Moreover, multi-agent system can be easily reconfigured by addition or deleting agents from the platform, without any influence on the rest of the platform (the other active agents), which allow us to deploy a new agent with new services any time, this gives the global system new features without the need to shut down or restart the system. A second platform can be deployed to migrate or duplicate all agents for a possible maintenance or update of the main platform, in order to insure the non-stop feature of the system.

Multi-agent systems can be developed by JACK [15], JADE [16], JADEX [17], etc. And the interaction between the agents of the various platforms is possible through the standards imposed by the FIPA [18], for this reason, the use of agents for dynamic reconfiguration approach does not problem of heterogeneity because the communication is standardized by the FIPA-ACL language [18], hence virtualization of MAS development technologies. This gives developers the freedom to choose their own development platform while ensuring interaction with other agents in the system, even if they are developed by different technologies. Therefore, the possibility to integrate new subsystems in the overall system.

In MAS, agent can be a smallest independent entity of the system for the following reasons:

- Each agent can be deployed independently.
- Each agent has his own behaviour.
- Each agent offers one or more services.
- Each agent uses the FIPA-ACL standard to communicate with other agents, whatever their development platform.

So the dynamic reconfiguration of the system based agent can be:

- Adding a new agent.
- Deleting an agent.
- The deployment of the services offered by the agent.

III. PROPOSED APPROACH

LPWAN is one of the systems which they need to reconfigure dynamically their pervasive system, and since they cover a large area, a large number of connected terminals will be present on the network, and each terminal represent a sub-System and offer several types of information in order to design a real world projection in the virtual world. This information can be a temperature, a pressure, a humidity rate, a movement, a speed of an object, etc.

The dynamic reconfiguration of the LPWAN becomes a major priority when the number of terminals increases,

in particular when new terminals appear in the network, in order to deal with the full new services offered by them. This is ensured by the pervasive system layer as shown in the Fig. 2, that must manage all the information coming from the terminals (sensors) and automatically reconfigured whenever a new terminal appears in the environment, for this an automatic mechanism is necessary to perform the reconfiguration of the global network.

Multi-agent systems can handle the complexity of solutions through decomposition, modelling and organizing the interrelationships between terminals. That was when we have modelled the terminal by an agent which has an access to the information offered by the terminal sensor, also it can publish this information as services to other agents of the platform.

Therefore, we propose a new approach to perform the dynamic reconfiguration which is based on the following agents:

- Terminal agent: It is implemented at the terminal which has as tasks:
 - Discover the environment of the terminal and signal its presence in the network.
 - Publish the services offered by the terminal.
 - Make services network more accessible and easily taken in.
- Receptionist Agent: It is implemented at the main platform which has as tasks:
 - Receive the new terminal agents, and make their services exploitable across the platform by polling the new agent using a messaging system.
 - Reconfigure the platform to make these new services visible to users.
- Yellow Pages Agent: It is implemented at the platform and which register all the services offered by the other agents, in the case of the JADE technology this agent is already implemented and known under the name The Directory Facilitator (df) [19].

Based on these three types of agent we proposed the following approach for the dynamic reconfiguration:

- The terminal agent automatically searches and locates the main platform.
- The terminal agent automatically connects to the platform.
- The terminal agent publish its services at the “df” agent.
- The “df” agent inform the receptionist agent for the arrival of a new terminal agent and its services.
- The receptionist agent automatically reconfigures the platform in order to take into account the services offered by the new terminal agent.

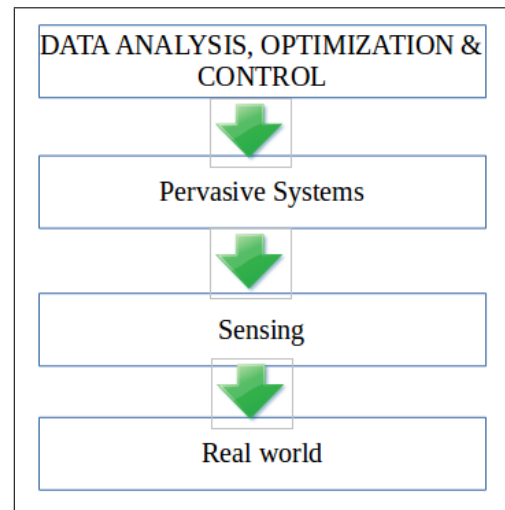


Fig. 2. The pervasive system layer in the data acquisition process.

IV. IMPLEMENTATION AND RESULTS

In the present approach, our solution is divided into two parts. The first one is implemented in the pervasive system layer of the LPWAN network (main platform) in order to get a better control of all the terminals that appears in the network, the second one is implemented in the terminal where will develop an agent (terminal agent) with the main behaviour is to locate the main platform in the network and publish its services.

Our solution is developed in JAVA using the JADE multi-agent systems development platform, the choice is made for the following reasons:

- JAVA: To ensure the portability of the solution on several kinds of terminals.
- JADE: It is a platform developed under JAVA, which implements yellow page agent which is called “df”.

The proposed solution is a kind of coordination between the receptionist agent of the main platform and the terminal agent; this is possible due to the different messages defined by the FIPA-ACL standard. These two agents plus the yellow page agent which is already defined in the JADE platform are cooperating in order to reconfigure the LPWAN pervasive system layer, by applying the proposed approach in the previous section and according to the scenario illustrated by Fig. 3:

A. Terminal Agent

This agent represents a node in the LPWAN network, physically it is a simple embedded system equipped with a sensor, an antenna to ensure the connection and a reduced operating system in which we will implement the terminal agent with access to the information provided by the sensor. The functioning of this terminal should consider the following steps:

- The connection to the network and obtaining IP address: This is possible because of the service offered by DHCP.

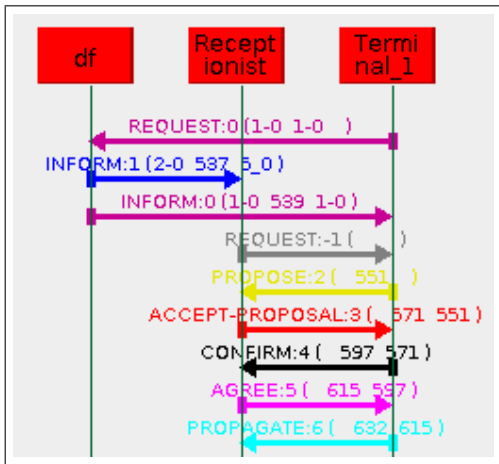


Fig. 3. Scenario for agent-based dynamic reconfiguration.

- Location and connection to the main platform.
- Services publication for the agent “df” of the main platform.
- Communication of the captured data by the sensor.

B. Receptionist Agent

This agent is implemented at the main platform, its main task is receiving the new terminal agents, this is possible through the “df” agent which will inform him with an ACL message (INFORM) every time an agent published its services in the platform, after the receptionist agent will interrogate the new terminal agent to get the different data captured by the latter and makes them accessible through the graphical interface included in the main platform.

C. Agents Communication

As shown in Fig. 3, the dynamic reconfiguration process is triggered by the arrival of a new terminal in the network, after that, a series of FIPA-ACL messages will be exchanged between the different agents defined by the approach according the following detailed scenario:

- The terminal agent sends a REQUEST message to the “df” to inform him of his presence and asks him to register his services in the platform.
- The agent “df” sends an INFORM message to the terminal agent to inform him that his request was accepted.
- The agent “df” sends another INFORM message to the receptionist agent to inform him of the presence of a new terminal agent and also communicates his services to him.
- The receptionist agent sends a REQUEST message to the terminal agent asking for the information offered by the services offered.
- The terminal agent sends a PROPOSE message to the receptionist agent for proposing the data format to send.

- The receptionist agent configures the main platform to receive the data from the new terminal agent and sends an ACCEPT-PROPOSAL message to the terminal agent informing him that his proposal was accepted.
- The terminal agent sends a CONFIRM message to the receptionist agent to indicate the start of sending data.
- The receptionist sends an AGREE message to the terminal agent to confirm that he is ready to receive the data.
- The terminal agent sends a PROPAGATE message to the receptionist agent to inform him of the sending of the first data and so on.

When the terminal agent disconnects the JADE platform automatically detects it and informs all agents through an INFORM message sent by The Agent Management System (AMS) [19], this allows the receptionist agent to reconfigure the system for the absence and breaking of data from the disconnected terminal agent.

V. DISCUSSIONS

Comparing our approach to previous approaches such as primary/backup system, component software or reuse embedded system resources; we will notice that the main difference is how an element of the system is detected. In our approach, it is up to the terminal agent to discover its environment and look for the main platform to publish its services. However, the other approaches leave this task for the global system, which further complicates the search and the discovery of all subsystems present in environment, particularly where their number becomes important.

Knowing that the agent is an independent entity which means that the addition or deletion of an agent have no influence on the overall functioning of the system. This implies a continuity of operation of the system without the services provided by the agent or the disconnected terminal. therefore, the number of bugs in the application is minimized and the no-stop of the system is ensured, because the addition or deletion of the terminal agent do not need to restart the system.

After this comparison, we can say that the proposed approach allows us to develop an application with two main functionalities:

- Accept new services offered by terminal agents.
- Reconfigure the main platform (application) to make new services used by user.

These two features allow the application to expand and update its services with new services offered by all terminals agent, which is a remarkable addition to the LPWAN pervasive system.

VI. CONCLUSION AND FUTURE SCOPE

The application of dynamic reconfiguration to the LPWAN pervasive system layer offers more flexibility to the network

and a better perception of its environment since the terminals represent a set of sensors and a better adaptation the various changes by adding or removing a terminal. Our approach ensures the following characteristics:

- Flexibility.
- Independence.
- No-stop.
- Modularity.

These four characteristics can be deployed in several kind of systems such as sensor networks, smart homes or smart cities, the cyber physical systems [20] and any telemetric application where the global system has a pervasive system layer with several independent components.

Therefore, the future scope of the current proposed approach can be developed as a Framework. Then we can just use this Framework in various applications such as the addition of new devices in computer systems. In this case our approach can enable the operating system to take into account all new devices without the need to install drivers; this is possible by developing the main platform in the operating system, and terminal agent in the device that aims to publish the services offered by the hardware and how to use it. This can give a new vision for the development of the computer peripheral.

REFERENCES

- [1] J. Petäjälä, K. Mikhaylov, M. Hämäläinen, and J. Iinatti, "Evaluation of lora lpwan technology for remote health and wellbeing monitoring," in *Medical Information and Communication Technology (ISMICT), 2016 10th International Symposium on*. IEEE, 2016, pp. 1–5.
- [2] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of lorawan," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.
- [3] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.
- [4] C. A. Iglesias, M. Garijo, and J. C. González, "A survey of agent-oriented methodologies," in *International Workshop on Agent Theories, Architectures, and Languages*. Springer, 1998, pp. 317–330.
- [5] W. Li, T. Logenthiran, and W. Woo, "Intelligent multi-agent system for smart home energy management," in *Innovative Smart Grid Technologies-Asia (ISGT ASIA), 2015 IEEE*. IEEE, 2015, pp. 1–6.
- [6] J. POLAKOVIC, "Architecture logicielle et outils pour systèmes d'exploitation reconfigurables," Ph.D. dissertation, INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE, 2011.
- [7] A. Shraer, B. Reed, D. Malkhi, and F. P. Junqueira, "Dynamic reconfiguration of primary/backup clusters," in *USENIX Annual Technical Conference*, 2012, pp. 425–437.
- [8] W.-T. Tsai, W. Song, R. Paul, Z. Cao, and H. Huang, "Services-oriented dynamic reconfiguration framework for dependable distributed computing," in *COMPSAC*, vol. 1, 2004, pp. 554–559.
- [9] S. H. Moon and C. sick Lee, "Dynamic management software design in embedded system using middle," 2014.
- [10] N. S. Voros and K. Masselos, *System level design of reconfigurable systems-on-chip*. Springer, 2005.
- [11] A. Taherkordi, P. Herrmann, J. O. Blech, and A. Fernandez, "Service virtualization for self-adaptation in mobile cyber-physical systems," in *International Conference on Service-Oriented Computing*. Springer, 2016, pp. 56–68.
- [12] H. Gao and H. Miao, "A quantitative model-based selection of web service reconfiguration," *2013 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp. 365–371, 2013.
- [13] M. Weiser, "The computer for the 21 st century," *Scientific american*, vol. 265, no. 3, pp. 94–105, 1991.
- [14] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *IEEE Personal communications*, vol. 8, no. 4, pp. 10–17, 2001.
- [15] M. Winikoff, "Jack intelligent agents: an industrial strength platform," in *Multi-Agent Programming*. Springer, 2005, pp. 175–193.
- [16] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing multi-agent systems with JADE*. John Wiley & Sons, 2007, vol. 7.
- [17] A. Pokahr, L. Braubach, and W. Lamersdorf, "Jadex: A bdi reasoning engine," in *Multi-agent programming*. Springer, 2005, pp. 149–174.
- [18] A. Fipa, "Fipa acl message structure specification," *Foundation for Intelligent Physical Agents*, <http://www.fipa.org/specs/fipa00061/SC00061G.html> (30.6. 2004), 2002.
- [19] F. Bellifemine, A. Poggi, and G. Rimassa, "Jade—a fipa-compliant agent framework," in *Proceedings of PAAM*, vol. 99, no. 97-108. London, 1999, p. 33.
- [20] S. Zanero, "Cyber-physical systems," *Computer*, vol. 50, no. 4, pp. 14–16, 2017.