Enhancing smart grid operation by using a WSAN for substation monitoring and control

Natalie Matta*†, Rana Rahim-Amoud†, Leila Merghem-Boulahia* and Akil Jrad†
*ICD/ERA (UMR CNRS 6279), Troyes University of Technology, France
Email: {natalie.matta, leila.merghem_boulahia}@utt.fr
†EDST, Centre-Azm, Laboratoire des Systèmes électroniques, Télécommunications et Réseaux (LaSTRe), Lebanese University, Lebanon, Email: {rana.rahim, ajrad}@ul.edu.lb

***Abstract*—One of the main preoccupations of the next generation power network, i.e. the smart grid, is improving the reliability of the power network. In order to achieve that, monitoring and control applications play an important role, in particular at a substation's level. Using a wireless sensor and actuator network is an attractive option for these applications. However, what is missing so far is that information collection should be complemented with the ability to transform significant data into actions in a decentralized manner. In this paper, we address this issue by adopting an agent-based approach. Autonomous agents are embodied in sensor nodes in order to evaluate the importance of collected data and choose an appropriate communication policy. In order to evaluate each phase of the proposed algorithm, different variants are compared with the legacy client/server scheme. Simulation results have shown better performance in terms of network congestion, energy consumption, and transmission delay.**

***Index Terms*—Wireless sensor networks, smart grids, substation monitoring, multi-agent systems, information management.**

I. Introduction

In the path of electricity from generation plants to consumer premises, transmission and distribution substations have many important functions, such as stepping the voltage levels up (for transmission over long distances) and down (for distribution to clients), regulating voltage, etc. It is thus important to monitor substations and voltage quality in order to detect faults. The overall objective is to improve the reliability of the grid and be able to avoid large-scale blackouts that until this day still affect millions of customers (e.g. the 2003 northeast blackout in the U.S.). If a fault should occur, service providers want to reduce both its duration and its scale. Monitoring and control of electric equipment is indeed one of the functionalities required in the future generation of power networks, the smart grid.

The smart grid embraces advanced information and communication technologies in order to improve the efficiency, reliability, and security of its components and services. In particular, wireless sensor and actuator networks (WSANs) are considered promising candidates for smart grid monitoring applications. WSANs are pervasive, able to cover large areas, and have a lower cost and more flexibility compared to wireline options. However, appropriate decentralized management of the information collected by the WSAN will be crucial in the smart grid for two main reasons. First, as identified by Oracle in their recent report [1], smart grid deployments are giving service providers access to new information. While this is needed in order to achieve the objectives of a more efficient and reliable power grid, it is expected to create a data deluge that will overwhelm service providers if not managed and analyzed properly. A part of the solution to this issue relies in decentralized data processing along the way from the sensors and other entities (e.g. smart meters) collecting it to the service providers' premises. Second, if faults are to be detected and their harmful impact controlled in a timely manner, the collected information needs to be translated into "actionable intelligence" [1].

It is thus clear that there is a need for a decentralized intelligent data management and processing in the smart grid. A multi-agent system (MAS), a paradigm derived from the distributed artificial intelligence field, is an attractive solution that can satisfy these requirements. In fact, an MAS is formed by a group of autonomous intelligent agents deployed in a certain environment and which are able to make decisions and act upon that environment. The value of an MAS approach in the power industry has been asserted in many settings, notably by the IEEE Power Engineering Society [2]. When adopted in a WSAN, an agent-based information management approach can satisfy the needs of decentralized management and control in a substation that we have identified. This paper presents and evaluates such an approach.

In the remainder of this paper, some related works are presented in section II. Section III exposes our agent-based approach for data management in a substation's WSAN. Simulation results are shown in section IV. We conclude and suggest future work in section V.

II. Related Work

Improving the reliability of the power network is a key objective for the smart grid. Today's electricity system still suffers from power outages and blackouts due to the lack of automated analysis and poor visibility of the utility provider over the grid [3]. Deploying WSANs will give the service provider the needed view by collecting information from the different subsystems of the grid.

The opportunities and importance of WSANs in the smart grid were put forward in many research studies such as [4]–[8]. In fact, traditional wired-based monitoring of electrical systems is expensive to install and requires constant maintenance of communication cables; whereas smart grid communications need to be scalable and pervasive. Wireless sensor networks (WSNs) present the advantages of being low-cost, pervasive, flexible, and rapidly deployed. Indeed, the use of WSNs can insure a wide coverage, even of remote sites, as well as a distributed and decentralized architecture which overcomes the issue of a single point of failure.

Practical implementations of WSNs for substation monitoring have been put in place. An example is a WSN of 122 nodes in a substation in Kentucky, U.S. [9]. In [10], a decentralized voltage quality monitoring architecture that employs wireless sensor networks is introduced. The advantages of such an architecture over client/server ones, are its scalability and task distribution which takes the load off the remote server. The authors consider a scenario where each part of the power grid is monitored by a sensor network, and propose a cooperation mechanism between these networks in order to propagate their respective quality indices. However, how the nodes in a single network communicate to evaluate the voltage quality index is not treated in [10].

MASs have also been applied to power quality monitoring. In [11], a conceptual MAS is proposed where agents are used to locally compute quality indices. The advantages of the provided agent architecture include adaptability, reconfiguration in case of failure, and the ability to assist operators in decisions related to power quality events. Nevertheless, the paper does not consider the underlying communication network, and only presents an MAS concept. The same issue is found in [12] where the authors apply an MAS in order to control in a decentralized manner a low voltage distribution network in which distributed generation and demand side management are included. Communications between the agents are not simulated over any networking technology, and the authors point out that they might become an issue when the system is scaled up.

In this paper, a WSAN is deployed in an electric substation. Each node is embodied with an agent, thus forming an MAS. Our main contribution is enabling the decentralized analysis of collected data and substation control using the WSAN. Our solution aims at monitoring electric components and characteristics so that faults can be predicted. Fault detection can be done in a precise and timely manner reducing as a result the response times to faults. In addition, when a fault occurs, it is important to take action as fast as possible in order to limit its extent (e.g. activating a circuit breaker to isolate a certain area so that the problem does not propagate). These points are exposed and further discussed in the following section.

III. WSAN for Substation Monitoring and Control

In this section, we present our proposed algorithm *Priority-Based Agent Cooperation (PriBAC)* for managing the information collected by a WSAN deployed at a substation. The contribution of our work lies in the ability to transform data into direct actions and to extract correlated information through decentralized analysis and agent cooperation. PriBAC is a priority-based communication scheme which employs two steps to achieve its objectives:

1) Data evaluation: consists of evaluating the priority associated with each detected value;
2) Communication policy selection: depending on the priority of the value, the appropriate communication policy is selected.

A. Data evaluation and policy selection

PriBAC currently defines three priority levels {Pri-1, Pri-2, Pri-3}, Pri-1 being the most important. Each priority is associated with a communication policy as shown in Table I.

TABLE I
Possible value priorities and their associated communication policy

Priority	Type	Communication policy
Pri-1	Fault	Priority-based value reporting
Pri-2	Warning	Agent cooperation
Pri-3	Normal	None

For each of the sensed parameters, PriBAC decomposes its range of values into intervals, and associates a priority with each of these intervals. When a node senses a given value *x*, the embodied agent will evaluate *x* by determining to which interval it belongs to, and thus determine its associated priority *p(x)*. For example, if a transformer's temperature rises above 55 degrees over ambient temperature by 10 to 15% then the detected temperature is a Pri-2 value. Any temperature reading that indicates an increase greater than 15% would be a Pri-1 value. Next, *p(x)* is tested. If it is equal to 1, it signals an event which should be reported to the sink (and eventually to the service provider's Supervisory Control and Data Acquisition (SCADA) system). Furthermore, if the node has knowledge about an actuator that should be alerted in that case, then a message is also sent to that actuator. PriBAC assigns the priority-based value reporting policy to Pri-1 values. This policy consists in sending the value to the sink, after processing it to determine its priority. Alternatively, in case of a Pri-2 value, PriBAC considers that the value should be reported because it may signal a fault or a problem, but that it is less urgent than Pri-1 values. In that case, PriBAC aims at defining the extent of the detected electric problem. The adopted policy in that case is the agent cooperation which will be explained in the next paragraph. Finally, Pri-3 values are considered normal operation values that do not need to be signaled. PriBAC simply keeps a record of these values, as well as of all values (of any priority), in the form of an average. These steps are presented in Algorithm 1.

Algorithm 1 Main PriBAC algorithm

$sensVal \leftarrow getSensValue()$
$p \leftarrow getPriority(sensVal)$
switch (p)
case 1**:**
  $updateAverage(sensVal)$
  if $(alert\_actuator = true)$ **then**
    $sendMessage(actuatorId)$
  end if
  $sendMessage(SINK)$
case 2**:**
  $updateAverage(sensVal)$
  $setTimer(POLL, waitTimeBeforePoll)$
case 3**:**
  $updateAverage(sensVal)$
end switch

B. Communication policy for Pri-2: agent cooperation

The agent cooperation policy adopted for Pri-2 values defines two phases: a polling phase, and an aggregation phase.

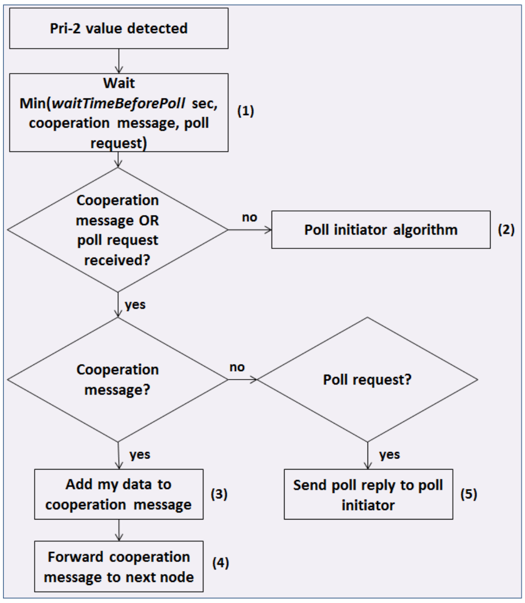

Fig. 1. Flowchart of the agent cooperation algorithm

When a sensor detects a Pri-2 value, it will wait for a pre-defined interval (represented by *waitTimeBeforePoll*) before initiating its *polling* phase (1)[1]. During this interval, if either a poll request or a cooperation message is received by the agent, it will respond accordingly. In the first case, when a poll request is received, the agent will respond by sending its detected value (5). In the second case, when getting a cooperation message, it will aggregate its data to that message (3), then forward the message to the next node on the way to the sink (4). Currently, PriBAC uses the average function for aggregation. The choice of the aggregation function can depend on the type of parameter considered. In addition, since many parameters are monitored in this environment, it is possible to define a correlation function between these parameters and use it for the aggregation (this will not be further detailed as it is out of the scope of this paper).

If the pre-defined interval expires and neither of the two previous actions occur, then the sensor agent initiates the *polling* mechanism (2). This agent is referred to as the *poll-initiator*. During this phase, the poll-initiator sends poll requests to the sensors in its coverage area asking them to cooperate (6), and waits for their replies, if any. Because neighbors may not always have data to send, PriBAC defines a *waitTimeForPollResponses* interval during which the poll-initiator waits for its neighbors'

[1]The numbers between parentheses refer to the steps of the flowcharts in Fig. 1 and 2.

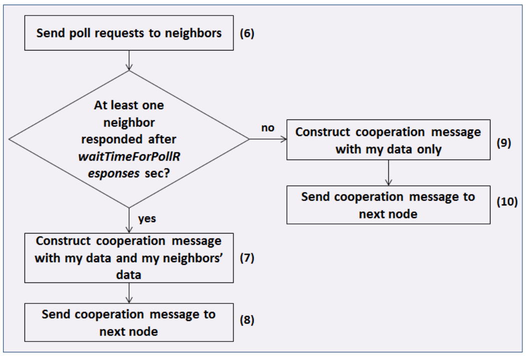

Fig. 2. Flowchart of the poll initiator algorithm

replies. If a neighbor has a Pri-2 value that it has not sent yet, it responds to the poll initiator by communicating that value. The initiator then constructs a cooperation message which contains its neighbors' received values, if any, aggregated with its own value, as well as the identification of all participating nodes (*nodeID*) (7,9). It then passes it to its one-hop neighbor on the way to the sink (8,10). This neighbor, having received a cooperation message, aggregates its Pri-2 value (which has not been sent yet) and its address to the message, and passes it to the next node on the way to the sink. The same is done at each node until the sink is reached.

When a cooperation message reaches the sink, it contains the aggregated values and IDs of the participating nodes. The sink can identify each node by its ID and so determine the first, last, and all nodes that participated in the cooperation message, allowing it to identify the extent of the problem. This is crucial when attempting to locate a fault and determining its extent and any damaged components.

The next section presents the performance evaluation study of PriBAC.

IV. Performance evaluation of PriBAC

Simulations were conducted using the Castalia simulator [13]. The main parameters used in the simulations are presented in Table II. Simulations are repeated 4 times. In a field of 30x12 meters, the network density is varied by considering networks between 40 and 150 nodes (this gives us higher densities compared with the deployment in [9]). Values are sampled each 4 ms. Radio parameters are chosen as those of the CC2420 transceiver [14] which equips a wide range of sensor nodes, such as the Tmote Sky [15]. We have used the T-MAC protocol implementation provided by the simulator as the MAC layer [16] [17]. On the routing level, multipath rings routing, implemented in Castalia (as a simplified version of [18]), is used. As its name indicates, multiple paths to the sink may exist, because nodes at a given level (a node's level is defined in terms of number of hops on the way to the sink) relay the messages of the nodes of the upper level. This ensures that messages can still be delivered even if some nodes experience failures. Finally, the sensors detect 23% of Pri-1 values and 46% of Pri-2 values in the network. These percentages are high and may not be realistic, but they are used to test the performance of our approach under drastic conditions.

TABLE II
Main simulation parameters

Parameter	Value
Number of repetitions	4
Field size	30x12 meters
Number of Nodes	40-150 nodes
Sampling Interval	4 ms
Radio	CC2420
MAC	T-MAC
Pri-1 values	23%
Pri-2 values	46%

A. Tuning PriBAC's parameters

In the first set of simulations presented here, we study the effect of *waitTimeBeforePoll* parameter on the performance of PriBAC. Note that:

- *waitTimeBeforePoll* is the amount of time that a node waits before it initiates a polling phase when it detects a Pri-2 value. This parameter is important in order to benefit the most out of the aggregation. In fact, if all nodes start polling their neighbors as soon as they detect such values, then too many cooperation messages will be sent. In addition, these messages will not carry much information. On the other hand, if a node waits too long to send its value (e.g. when *waitTimeBeforePoll* is greater than the sampling interval), it may be missing other important values.
- *waitTimeForPollResponses* is the time interval during which a poll-initiator waits for the responses of its neighbors to its poll request. In the following simulations, *waitTimeForPollResponses* is set to 0.1 μs.

In order to evaluate the behavior of PriBAC when this parameter takes values of different orders, *waitTimeBeforePoll* is varied and takes the values 0.001, 0.01, 0.1, 1, and 10 ms. The results obtained are very similar for 0.001, 0.01, 0.1, and 1 ms. Figure 3 shows the average number of cooperation messages sent per node. We can

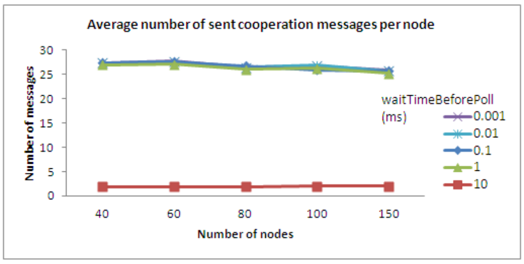

Fig. 3. Average number of cooperation messages sent per node

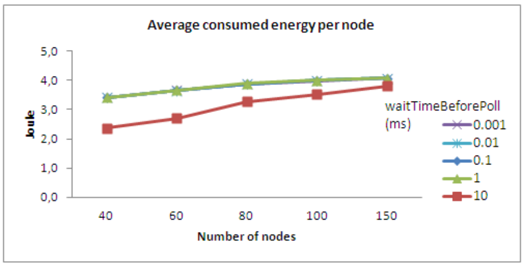

Fig. 4. Average consumed energy per node

waitTimeBeforePoll (ms)	0.001	0.01	0.1	1
Percentage of cooperating nodes	60.1	66.7	67.6	65.8

Fig. 5. Percentage of cooperating nodes

see that as the timer's value increases to 10 ms, less cooperation messages are sent resulting in less energy consumption per node as shown in Fig. 4 (Note that even with an average of around 27 cooperation messages sent per node, PriBAC significantly reduces the number of sent messages per node, as it will be shown in Fig. 6). However, 10 ms is not a good value for this timer, as it is greater than the sampling interval (4 ms). If a node waits more than 4 ms to send a cooperation message, then PriBAC will not be able to guarantee data accuracy.

Having eliminated the 10 ms value, we take a look at the remaining values. Figures 3 and 4 show very similar results for these values of the *waitTimeBeforePoll* timer. Looking at the percentage of nodes participating in an aggregation session (Fig. 5), 0.1 ms achieves the highest percentage and thus favours the cooperation more than the other values. In the second set of simulations, *waitTimeBeforePoll* is equal to 0.1 ms.

B. Comparison with other approaches

In the second set of simulations, PriBAC is compared with other approaches in order to evaluate the performance of each of its phases. These approaches are:

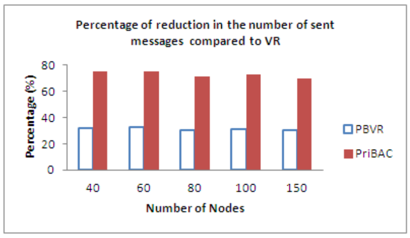

Fig. 6. Percentage of reduction in the number of sent messages with respect to VR

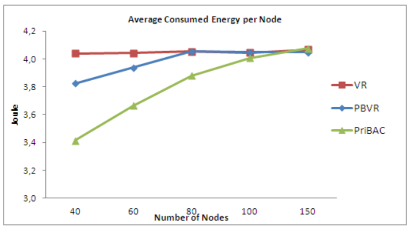

Fig. 7. Comparison of average consumed energy per node

- Value Reporting (VR): sends all sampled values to the sink without any processing. It corresponds to the classic client/server scheme.
- Priority-Based Value Reporting (PBVR): sends only Pri-1 and Pri-2 values to the sink. We have proposed this scheme as an intermediate between the two other approaches.

Figure 6 shows the percentage of reduction in the number of sent messages with respect to the VR approach. The results clearly show that both approaches significantly reduce the number of sent messages in the network. This is due to the priority-based evaluation phase which selects important information that should be transmitted. Furthermore, we have noted that PriBAC achieves an average gain of 66% compared to PBVR. That means that PriBAC uses less messages to transmit the same information, which shows the benefit of the agent cooperation's aggregation phase of PriBAC. These results also reflect on the average consumed energy per node as shown in Fig. 7. In fact, sending less messages results in consuming less energy. Note that as the number of nodes increases, the number of messages received by a node increases more rapidly (as it is receiving messages from a higher number of nodes). This growth in the number of exchanged messages reflects on the consumed energy per node for each of the compared approaches.

Next, we look at the transmission delay of application messages. We compare the average delay of sent messages for PriBAC, VR, and PBVR approaches in Fig. 8.

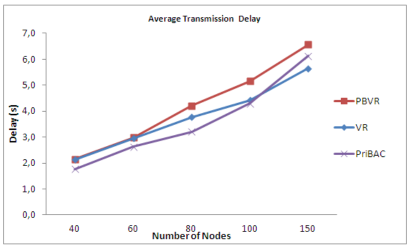

Fig. 8. Comparison of average transmission delay

The delay increases with the number of nodes, which is a consequence of the increased number of exchanged messages with respect to the number of nodes. For a network of up to 100 nodes, PriBAC achieves better transmission delay performances than VR and PBVR. In a network of 150 nodes, VR has the least transmission delay. This is expected because PriBAC requires intermediate nodes to do some processing, as opposed to the two other approaches where no processing is done at intermediate nodes. When the number of nodes increases, the average delay for PriBAC messages witnesses a faster growth as a result of this processing along the way to the sink. We expect that reducing the processing at the application layer will mitigate this effect.

Finally, it should be noted that PriBAC achieved these performances under drastic conditions where high percentages of important data are detected in the network. On one hand, it has succeeded in reducing the energy consumption of the sensor nodes and thus prolonging their lifetime. On the other hand, PriBAC has proven to reduce network traffic and detect faults faster, which in turn will reduce the quantity of information that needs to be communicated by the sink to the utility provider, thus requiring less bandwidth, less storage, and less treatment on the utility's side.

V. Conclusion

This paper has identified a key requirement of the next generation power grid, which is the decentralized monitoring and control of electric substations and their components via a WSAN. The need for a smart information management of the data collected by that WSAN was addressed in this work. We proposed the PriBAC algorithm to reduce energy consumption in sensor nodes, and intelligently manage sensed data so that values can be prioritized and that faulty components can be identified. PriBAC adopts a priority-based agent cooperation approach. Simulation results have shown that PriBAC performed better in terms of energy consumption, number of sent messages and delay, compared to the other approaches. In the future, we plan to include an evaluation of how PriBAC affects the functionality of the substation (e.g. response time to faults), and compare it to other cooperative solutions. In addition, instead of having fixed intervals, fuzzy logic could be used for determining priority intervals.

Acknowledgment

This work was supported in part through grants from the Troyes University of Technology and the Lebanese University.

References

[1] Oracle, "Big data, bigger opportunities: Plans and preparedness for the data deluge," *Technical report*, Jul. 2012.

[2] S. McArthur *et al.*, "Multi-agent systems for power engineering applications;part i: Concepts, approaches, and technical challenges," *Power Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 1743 –1752, nov. 2007.

[3] L. S. C. for the U.S. Department of Energy, "The smart grid: an introduction," *Technical report*, Oct. 2008.

[4] B. Bilgin *et al.*, "On the performance of multi-channel wireless sensor networks in smart grid environments," in *Computer Communications and Networks, 20th International Conference on*, 2011, pp. 1 –6.

[5] S. Isaac *et al.*, "A survey of wireless sensor network applications from a power utility's distribution perspective," in *AFRICON*, sept. 2011, pp. 1 –5.

[6] V. Gungor *et al.*, "Opportunities and challenges of wireless sensor networks in smart grid," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 10, pp. 3557 –3564, oct. 2010.

[7] S. Ullo *et al.*, "The role of pervasive and cooperative sensor networks in smart grids communication," in *15th IEEE Mediterranean Electrotechnical Conference*, april 2010, pp. 443 –447.

[8] M. Erol-Kantarci *et al.*, "Wireless multimedia sensor and actor networks for the next generation power grid," *Ad Hoc Networks*, vol. 9, no. 4, pp. 542 – 551, 2011.

[9] A. Nasipuri *et al.*, "Design considerations for a large-scale wireless sensor network for substation monitoring," in *Local Computer Networks, 2010 IEEE Conference on*, pp. 866 –873.

[10] M. di Bisceglie *et al.*, "Cooperative sensor networks for voltage quality monitoring in smart grids," in *PowerTech, IEEE*, 2009.

[11] S. Srivastava *et al.*, "A conceptual power quality monitoring technique based on multi-agent systems," in *37th Annual North American Power Symposium*, 2005, pp. 358 – 363.

[12] G. Hutchinson *et al.*, "A multi-agent system for decentralised control of low voltage distribution networks," in *Universities Power Engineering Conference, International*, 2010, pp. 1 –6.

[13] "Castalia simulator website," http://castalia.npc.nicta.com.au/.

[14] T. Instruments. cc2420 product description. http://www.ti.com/product/cc2420#description.

[15] Moteiv, "Tmote sky brochure," http://sentilla.com/files/pdf/eol/tmote-sky-brochure.pdf.

[16] T. van Dam *et al.*, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003.

[17] Y. Tselishchev *et al.*, "Experiences and lessons from implementing a wireless sensor network mac protocol in the castalia simulator," in *Wireless Communications and Networking Conference, 2010 IEEE*, april 2010.

[18] S. Nath *et al.*, "Synopsis diffusion for robust aggregation in sensor networks," *ACM Trans. Sen. Netw.*, vol. 4, no. 2, Apr. 2008.