Wireless Sensor Network Architecture based on Fog Computing*

Viorel Mihai, Cristian Dragana, *Member IEEE*, Grigore Stamatescu, *Member IEEE*, Dan Popescu,
Member IEEE, and Loretta Ichim, *Member IEEE*

Abstract— **Wireless Sensor Network (WSN) has been a focus for research in the last years due to the promising technology it embeds. This appears to be the most sustainable technology for environmental sensing whether it's about limited or large-scale monitoring, thanks to the ad-hoc wireless links, scalability and ease of implementation. However, main drawbacks are stemming from the limited capacity of network nodes for data storage, computing and accessing. To overcome these limitations, virtualized resources were appended allowing access to increased storage, processing and user-friendly accessibility. This came as a natural development of the common WSN architectures in the trend of modern concepts emerged with the IoT (Internet of Things) technologies proliferation. Despite the increasing usage of cloud-based WSN monitoring systems, there are still issues due to the drawbacks of cloud computing such as latency and storage costs. This paper discusses the improvements made to a cloud-based WSN architecture by adding a layer of computing at the edge of the network, a method that follows the novel model of analysing and acting on IoT data, entitled Fog Computing. Comparative analytics were performed to prove the improvements achieved through edge of the network computing.**

I. INTRODUCTION

The IoT concept is generally characterized by large volume of interconnected devices, widely distributed. These devices have low process capabilities and limited data storage [1]. Furthermore, an emergently necessity is make sense of all the data gather from multiple such devices. The technology that fits these needs is the cloud computing. Cloud computing is massive adopted in IoT scenarios due to potential unlimited computing power and data storage. It offers a centralize solution for statistical data analysis, historical data storage and data visualization capabilities.

Regarding to IoT implementations, scalabilities problems may occur due to high network density. A noticeable necessity is to reduce data flow at local level and transmit towards cloud platform just the relevant data [2].

The delay introduced by high among of data transferred to the Cloud, most often provides irrelevant and redundant information. Furthermore, sending irrelevant data to the cloud for processing and storage, can saturate network bandwidth and compromise the entire application.

Fog computing is an emerging technology that proposes intelligent connection networks between IoT devices on one hand and cloud platforms on the other hand. Fog computing is an extension of cloud computing at the edge of network [3]. It provides local data aggregation and focuses on removing irrelevant data from network. The focus of this layer is to provide an open, distributed, scalable and secure IoT communication topology towards cloud platform.

The outline of the paper is as follows: Section II provides a brief overview of related work. Section III describes the initial communication architecture and added Fog computing layer. Section IV describes the results and the improvements made by adding Fog computing capabilities at local processing unit. In the last chapter, we synthesize the conclusion of this paper.

II. RELATED WORKS

A novel cloud-based system architecture was proposed in [4], tailored for process monitoring, data analysis and statistical study of network performance. Advantages and challenges were discussed with focus on an industrial wireless sensor network (IWSN) integration. A central coordinator node handles every interaction between the IWSN and the outside world, thus it enables the integration of large scale monitoring systems into reliable and resilient clouds for data acquisition, processing and storage.

In [5] the authors discuss the terminology and current definitions of fog computing, providing a more comprehensible definition of this concept. In a simplified form, we define the fog computing as a geographically distributed computing architecture, composed of heterogeneous connected devices (including edge devices) at the edge of the network and not exclusively supported by cloud services. References [6], [7], [8] and [9] discuss the benefits gain from fog computing.

In [10] the authors propose a gateway based fog computing architecture for WSAN (Wireless Sensors and Actuator Networks), consisting of sets of gateway and micro server, with different wireless network types such as Bluetooth Low Energy (BLE) and ZigBee. The connection between gateway and micro server is wired, using Ethernet interfaces, while gateways interconnection is achieved through wired and wireless communication (3G, Long-Term Evolution - LTE). This architecture provides an event driven virtualization model. A more specific application is presented in [11], where the authors discuss the architecture for connected vehicles with Road Side Units (RSUs) and M2M gateways based on the Fog Computing concepts. This prototype architecture deploys the Fog services at the RSUs and M2M gateways, enabling custom services such as M2M data analytics, management of data and connected devices. In

---

*Research supported by UEFISCDI, Bridge Grant Program, project SIMUL, BG 49/2016 and by ROSA Agency, Program STAR, project MAARS 185/2017.

V. Mihai, C. Dragana, G. Stamatescu, D. Popescu, and L. Ichim are with Department of Automatic Control and Industrial Informatics, University Politehnica of Bucharest, 060042 Bucharest, ROMANIA (e-mail: mihai.tc83@gmail.com; e-mail: c.dragana@imtt.pub.ro; e-mail: grig@me.com; e-mail: dan.popescu@upb.ro; e-mail: loretta.ichim@upb.ro).

[12] the authors propose a high-level programming model for large scale geospatially distributed applications entitled Mobile Fog. Model evaluation is driven through vehicle tracking using cameras and traffic monitoring applications. Moreover, different simulations were conducted using OMNeT++ [13], with realistic traffic patterns generated by SUMO [14].

The structure of fog computing layer is lacking in standardization. A way through standardization is paved by a consortium named *OpenFog* [15]. It proposes a comprehensive description about basic direction for this emerging domain. Seven pillars of fog computing are synthesized: *Security, Scalability, Open, Autonomy, RAS (Reliability, Availability, Serviceability), Agility, Hierarchy, Programmability.* The security pillar is based on encryption and hardware-based immutable root of trust. Scalability pillar is a property which describes the vertical and horizontal fog network and the ease to integrate new devices. Open pillar enables pooling discovery, a method to dynamically deploy fog nodes. Autonomy pillar support distributed decision making through network hierarchy. RAS pillar ensures a synergy connection between hardware, software and operations. Agility pillar support local computing transformation of data into information. Hierarchy pillar describes the network as smart devices grouped on physical or logical layers. Programmability pillar ensures ease of deployment at software and hardware level supporting dynamically node re-tasking.

Security and privacy issues are discussed in [16] according to the Fog computing paradigm. Typical attacks, such as man-in-the-middle attack, are investigated by examining CPU and memory consumption on Fog devices.

III. METHODS AND MATERIALS

Our work has brought improvements to a common cloud-based WSN system, developed for indoor air quality monitoring. The system provides a scalable architecture in which unprocessed row data is gather in a cloud environment. The system is deployed as in Fig. 1.

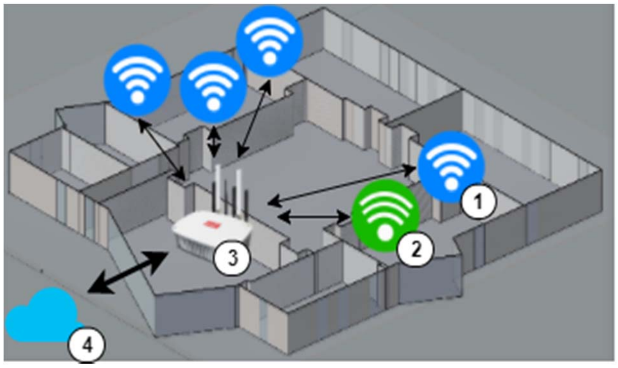

Figure 1. Wireless sensor network deployment: 1 – traditional wireless node; 2 – wireless node with fog computing capabilites, 3 – local gateway, 4 – cloud platform.

It can be considered as a sample of traditional cloud computing in which all data aggregation is done at superior level. We consider a specific implementation for a case scenario application. The architecture comprises five wireless nodes and one gateway, interconnected by ZigBee communication protocol. As cloud platform, Microsoft Azure [17] is used.

By considering all constrains of existing implementations, fog computing capabilities where added at node level, in a subset of the total node group. Each node can measure 6 air quality specific parameters: *Temperature, Humidity, CO2, NO2, VOC and CO.* At node level, local computing and communication capabilities are available.

In Fig. 1 two types of nodes are displayed. Blue nodes provide air quality measured values with a constant sampling rate. At each timestamp, the sensors are activated, physical values are measured. The ZigBee message is structured and sent towards local gateway. The green node is equipped with fog computing capabilities. As one can see, the system homogeneity was not alternated. The green sensor provides specific information about the air quality, using a local fog computing layer described in the next chapter.

Due to short distances between nodes and gateway, a star communication topology is used. Every node transmits wireless a ZigBee data packet to the local gateway. ZigBee communication protocol is used due to low-bandwidth, short-range and low-rate needs. By providing a low-cost communication protocol the system is enhanced with easy scalability proprieties and high-level data interchange. The local gateway collects data and manages data flow towards the cloud platform. Data storage and web server capabilities are available to support local information access.

The purpose of this article is to reduce the data flow from sensor to cloud by introducing a local fog computing layer. The solution is considering all the constraints of the existing implementation by enhance with fog computing capabilities a subset of all nodes without compromising homogeneity. Data flow is reduced by decreasing the number of transmitted data packets and the dynamic structural allocation of the ZigBee message. The improved architecture is presented in Fig. 2.

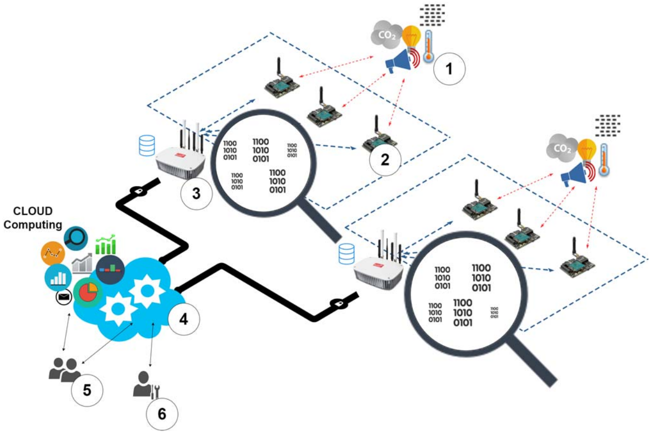

Figure 2. System architecture: 1 – air quality measured values; 2 – wireless node with fog computing capabilities; 3 – local gateways; 4 – cloud platform; 5 – users; 6 – administrator.

This architecture can be scaled at local level by increasing the number of devices connected to local gateway, but also by adding new gateways. The number of messages is decreased by deciding at local level if the message needs to be send. Also, the message length is decreased by dynamic allocation of the structure. By this, we refer at the node capability to decide if the measured data is relevant for the system. If so, the value is added in the ZigBee message. Basic data aggregation procedures were implemented at local fog computing layer to determine data relevance.

The first procedure triggers an event (E_1) when the investigated measurement exceeds a threshold value.

$$E_1 = \{e(v_i) \in Q | v_i > Y\} \tag{1}$$

where:

- E_1 is a Boolean value which determines if the measured value is included in the structure of ZigBee message if returned value is true;
- Q is the event set, $Q = \{E_1, E_2, E_3\}$;
- v_i is the measured value at iteration i;
- Y is the threshold value for E_1.

The second procedure (E_2) triggers an event when the difference between the last value and the current value is grater then an establish threshold.

$$E_2 = \{e(v_i, v_j) \in Q \big| \; |v_i - v_j| > Z\} \tag{2}$$

where:

- E_2 is a Boolean value which determines if the measured value is included in the structure of ZigBee message if returned value is true.
- Q is the event set, Q = $\{E_1, E_2, E_3\}$;
- v_i is the measured value at iteration i
- v_j is the measured value at iteration j, $j=i+1$
- Z is the threshold value for E_2

The last procedure triggers an event (E_3) when the investigated value exceeds an established bandwidth around sliding average value (*as*). It is defined as:

$$E_3 = \{e(v_i) \in Q | T_1 > v_i > T_2\} \tag{3}$$

where:

- E_3 is a Boolean value which determines if the measured value is included in the structure of ZigBee message if returned value is true;
- Q is the event set. $Q = \{E_1, E_2, E_3\}$;
- v_i is the measured value at iteration I;
- T_1 and T_2 are the threshold defines as in equation (4) and (5).

Thresholds are the following:

$$T_1 = as_i - p * as_i \tag{4}$$

$$T_2 = as_i + p * as_i \tag{5}$$

where:

- p is a percent that determines the bandwidth; in experiments p was considered 5%.
- as_i is the sliding average calculated at iteration i, defined in equation (6):

$$as_i = \frac{1}{n}\sum_{j=i-n}^{i-1} v_j \tag{6}$$

where:

- n is the range; in experiments n was considered 10
- vj is the measured value at iteration j

Considering this data aggregation procedures, at node level the following steps are followed. First, current physical values are measured. After that, the node decides if a measured value is relevant – considering implemented event detection procedures. If at least one event is generated, the node begins to structure a ZigBee message. The message is dynamically allocated with the measured values that generated an event. The pseudocode of fog computing is displayed in Algorithm 1.

Algorithm 1: Low scale data aggregator

```
Input: Time-stamp
Output: ZigBee message
Initialize:
  ToAdd[5];
for each Time-stamp
  Sensors.ON // enable sensors
  wait sensor_timeResponse;
  read sensors;
  Sensors.OFF; //disable sensors
  for each sensor
      if procedureGeneratesEvents
        ToAdd[sensor] = true
      end if
  end for
  if at least ToAdd[sensor] = true
      Initiate ZigBee_Message
  for each sensor
      if ToAdd[sensor] = true
        ADD value[sensor] //add sensor value to ZigBee message
      end if
  end for
```

To test this architecture, a comparative analyze was developed in a hybrid implementation which consists in normal nodes and nodes with fog computing capabilities. The results are mentioned in the next chapter.

IV. EXPERIMENTAL RESULTS

The original implementation was adapted to conduct the experiments. From a total of 5 wireless nodes, we configured one with the proposed local scale data aggregator algorithm. The data packet is dynamically structured. The maximum length is achieved when the evaluation procedures trigger an event for every measured value (in total 6 values). The rest of 4 nodes send data at a fix timestamp, as in original implementation. In this case, the packet structure is fixed and includes all measured data. The displayed result considers a comparative analyze of the data packets sent by one original node and one improved node. These packages were captured at gateway level. As cloud platform, Microsoft Azure was used. One snapshot of data packets is presented in Table 1 and in Fig. 3 in a graphic format.

As one can see, in the data packages snapshot from Table 1, the message from node 1 has always the same structure. By observing the value gather for each investigated measurement, it is obvious that some data are trivial payload. The improvement made by locally computing raw data are observable in the structure of the message from node 2.

TABLE I. DATA PACKAGES SNAPSHOT

t_i h:min:s	**E_i**	**N1**	**N2**	**T** C	**H** %	**CO_2** ppm	**CO** ppm	**NO_2** ppm	**VOC** ppm
…									
00:30:03		X		25.6	55.5	1.73	1.72	0.15	0.91
00:30:41		X		26.4	54.5	1.54	1.85	0.15	0.92
00:31:19		X		24.5	52.9	1.54	1.86	0.16	0.86
…									
01:35.17		X		24.2	47.7	1.66	1.80	0.17	0.86
01:35:40	**E_3**		**X**	**29.6**	**49.8**	-	-	-	-
01:35:55		X		24.1	47.4	1.64	1.72	0.16	0.86
01:36:34		X		26.1	48.2	1.65	1.70	0.17	0.86
…									
01:53:11		X		26.7	46.9	1.85	1.79	0.16	0.86
01:53:18	**E_3**		**X**	**22.9**	-	-	-	-	-
01:53:50		X		25.1	50.3	1.85	1.74	0.16	0.87
01:53:57	**E_3**		**X**	**29.6**	-	-	-	-	-
01:54:28		X		24.5	51.3	1.85	1.79	0.19	0.87
01:55:06		X		26.4	49.1	1.74	1.72	0.18	0.87
…									

Firstly, the ZigBee message is transmitted whenever at node level, for one measured data at least one procedure generates an event. Secondly, the message has a compressed structure which includes the relevant measured data.

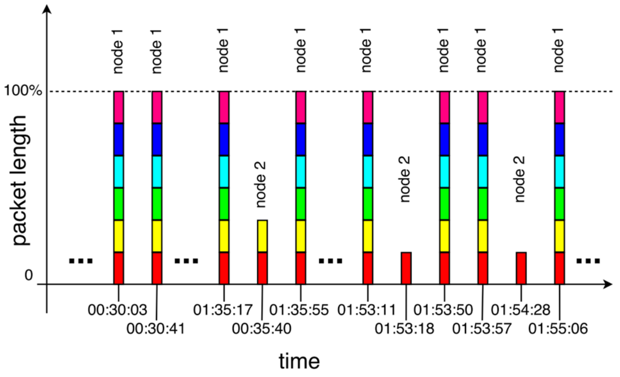

Figure 3. Received data packets at gatway level.

Only the relevant data are included in the ZigBee message which brings addition information at cloud level concerning the air quality. Further, some experiments were conducted to determine the field of use of each procedure. The result suggests that the E_2 procedure offers the best performance when is needed a relatively high data flow but still better results.

In Fig. 3 one can observe the messages rate in a given timeframe. It can be observed that the second message structure is more compact due to dynamical allocation. Further we consider a timeframe which include 1567 messages received from both nodes. Table 2 quantifies the statistical result of our experiments.

TABLE II. RECEIVED DATA PACKETS AT GATEWAY LEVEL

Total data packages received	1567
Node 1 total data packages	77.444 %
Node 2 total data packages	22.556 %
Node 1 average data packet length	100 %
Node 2 average data packet length	13%

The last experiment synthesizes the result in a numeric manner. From a total 1567 received message at cloud level, 1212 messages arrive from node 1 and 353 from node 2. The average message length for node 1 is equal with the maximum value due to static allocation. On the other hand, for node 2, the average message length is 13% from the maximum value. Also, on the run experiments, no maximum length message from node 2 were received.

V. CONCLUSION

In this paper we have proposed a local computing data aggregator algorithm to decrease the total amount of data send towards cloud platform. As a fog computing layer, we have adapted local computing algorithm in an existing cloud-based architecture, considering all the implicit constrains. We have obtained a heterogeneous solution consisting of wireless measure nodes with and without fog computing capabilities, local gateway and cloud platform.

At local level, this paper describes a basic data aggregator algorithm which allows local data fusion. Three procedures are considered. Each procedure triggers an event when a measured value reaches certain threshold values. These events enhance the local computing power with decision making capabilities. In these terms, the wireless node can decide if is necessary to send a ZigBee message. If so, it will decide which measured value contains relevant data air quality information.

The results show the improvements achieved through leveraging fog computing in a traditional cloud-based system. Considering fog computing, a large volume of redundant data is reduced. The benefits are multiple. Some worth mentioned are as follows. By decongestion of data traffic, network scalability is improved. In this case, low bandwidth communication protocol can be used to reduce implementation cost. Local energy consumption is reduced due to shorter transmission time. In [18] it is demonstrated that at node level, the costliest period in energy terms is sending and receiving data.

As future work, our focus is to develop more complex low-level data aggregator algorithm suitable for fog computing. One improvement on suggested algorithm is dynamically computation of threshold values. Also, by constantly adjusting the procedure used for every measured value the result will improve.

Field of relevance of such improvements can be included in recent areas of interest. In a holistic approach smart homing can utilize this low data communication framework to manage a greater number of row data sources. Our future efforts are focused towards health-related smart home automation. In this manner, bio-signals gathered from wearable sensors are integrated in our local data aggregator. Similar algorithms are used to generate event triggered alarms to monitor the health status of house's inhabitants. Using fog and cloud computing this information will be available at grater scale to specialists and multiple artificial intelligence algorithms that can provide a better health support without compromising private regulations. Such interest is present in up to date scientific articles. In [19] the authors present a fog-assisted system that allows a better healthcare in common circumstances. As future work we will correlate health specific data with air quality measurements to offer an improved environment.

REFERENCES

[1] A. Botta, W. de Donato, V. Persico, and A. Perscape, “Integration of cloud computing and Internet of Things: A survey,” *Future Generation Computer Systems,* vol. 56, pp. 684–700, 2016.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the Internet of Things” in *MCC '12 Proceedings of the first edition of the MCC workshop on Mobile cloud computing,* pp. 13-16, 2012.

[3] A. V. Dastjerdi, and R. Buyya, “Fog computing: Helping the Internet of Things realize its potential,” *Computer*, vol. 49, pp. 112-116, Aug. 2016.

[4] O. Chenaru, G. Stamatescu, I. Stamatescu, and D. Popescu, “Towards cloud integration for industrial wireless sensor network systems,” in *Proceedings of the 9-th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, pp. 917-922, 2015

[5] S. Yi, Z. Hao, Z. Qin, and Q. Li, “Fog computing: Platform and applications,” in *Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, pp. 1-6, Nov. 2015.

[6] A. Yousefpour , G. Ishigaki , and J. P. Jue, “Fog computing: towards minimizing delay in the Internet of Things,” *IEEE International Conference on Edge Computing* , June 2017.

[7] G. Ravikumar, N. Saikiran, and A. Sathish, “FOG: A novel approach for adapting IoT/IoE in cloud environment,” *International Journal of Engineering Trends and Technology (IJETT)*, vol. 42, pp. 189-192, Dec. 2016.

[8] S. Yi, C. Li, and Q. Li, “A survey of fog computing: concepts, applications and issues,” *Mobidata@MobiHoc*, pp. 1-6, 2015.

[9] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwalder, and B. Koldehofe, “Opportunistic spatio-temporal event processing for mobile situ- ation awareness,” in *Proceedings of the ACM international conference on Distributed event-based systems*, pp. 192-206, 2013.

[10] W. Lee, K. Nam, H.-G. Roh, and S.-H. Kim, “A gateway based fog computing architecture for wireless sensors and actuator networks,” in *20th IEEE International Conference on Advanced Communications Technology*, pp. 210–213, 2016.

[11] S. K. Datta, C. Bonnet, and J. Haerri, “Fog computing architecture to enable consumer centric Internet of Things services,” in *2015 IEEE International Symposium on Consumer Electronics*, pp. 1–2, June 2015.

[12] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwalder, and B. Koldehofe, “Mobile fog: A programming model for large-scale applications on the internet of things,” in *MCC’13 Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, pp. 15-20, 2013.

[13] A. Varga, and R. Hornig, “An overview of the OMNeT++ simulation environment,” in *Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, 2008.

[14] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “SUMO - Simulation of Urban MObility: An overview,” in *Advances in System Simulation*, 2011.

[15] I. Stojmenovic, and S. Wen, “The fog computing paradigm: scenarios and security issues,” in *Proceedings of the Federated Conference on Computer Science and Information Systems*, vol. 2, pp. 1-8, 2014.

[16] https://www.openfogconsortium.org/, 23.12.2018.

[17] https://azure.microsoft.com/en-us/, 23.12.2018.

[18] C. Dragana, G. Stamatescu, V. Mihai, and D. Popescu, “Evaluation of cluster formation algorithm in large scape wireless sensor networks”, in *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pp 1-6, September 21-23, 2017.

[19] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, “Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach,” *Future Generation Computer Systems*, vol. 78, pp. 641-658, 2018.