Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 151 (2019) 345–352

Procedia Computer Science

www.elsevier.com/locate/procedia

The 10th International Conference on Ambient Systems, Networks and Technologies (ANT)
April 29 – May 2, 2019, Leuven, Belgium

Sensor virtualization Middleware design for Ambient Assisted Living based on the Priority packet processing

Zubair Khalid[a], Usman Khalid[b], Mohd Adib Sarijari[c], Hashim Safdar[a], Rahat Ullah[a], Mohsin Qureshi[a] and Shafiq Ur Rehman[a]

[a]Department of Electrical Engineering, Federal Urdu University of Arts Science and Technology G7/1-Islamabad, Pakistan
bDepartment of Electrical Engineering, COMSATS Institute of Information Technology, Abboattabad, Pakistan
[C]Faculty of Electrical Engineering, Universiti Teknologi Malaysia (UTM) Johor, Malaysia

Abstract

Wireless Sensor Network (WSN) is leading towards a new and more advanced paradigm that is Internet of Everything (IoE). Ambient Assisted Living (AAL), which aims to support elderly and physically challenged persons, is based on sensor networks. WSNs are mostly installed for a particular single application. Though, future of WSNs hides in WSN virtualization, which supports multiple applications. The aggregation of the sensor network resources in the form of software or hardware are supported by WSN virtualization. In Virtualizing WSN, the role of middleware layer is of supreme importance. However, a middleware layer faces overhead challenges of high processing time, delay, and the power consumption. The major challenge faced by AAL application in the paradigm of WSN virtualization is the delay caused by the aggregation of the sensed data. In this research paper efforts have been made to design a priority based multi-threading middleware design for sensor virtualization. The proposed middleware resides at the sensor node as well as the gateway that reduces the delay caused by the sensor virtualization. The proposed middleware design has been implemented and tested on the AAL testbed.

Keywords: AAL; Sensor Virtualization; Middleware layer; sensing nodes.

* Corresponding author. Tel.: +92-321-9812353.
E-mail address: engr.zktanoli@gmail.com

1877-0509
10.1016/j.procs.2019.04.048

1. INTRODUCTION

In the recent past the researcher trends have moved from the smart cities and homes to smart universe. Where every household item is linked to one another [1, 2]. The trends and predictions show that the future of globe lies in the electrical technologies dealing in the progressively more interconnected world. We are moving from IoE [3], which deals in large federated WSNs that are to be connected seamlessly to one another. Sensing devices having squat data rate are immensely connected to Internet in large numbers. Approximately, 50 billion devices will be connected to the internet by the year 2020 and these figures show that the number connected devices will exceed the population of earth [4]. Advancement in agriculture, medicine, transportation, technology and industrial revolution has raised the living standards of life. These are one of the main factors that add up to a prolonged and the healthier living, when compared to the prior age group. Approximately, twenty percent of the total world's population will be aged sixty years or above by 2050 [5, 6]. AAL can facilitate many elderly people who are facing the challenges regarding health primarily due to their cognitive decay, vision and hearing loss, and the lack of physical strength. In U.S.A about 80% of citizens over the age of 65 years are living with at least one chronic illness [5]. AAL contribute in security, ventilating, health monitoring, air conditioning, and supporting elderly and physically challenged at home as revealed in Fig. 1.

In the paradigm of AAL large number of sensing devices monitor and continually assess the health and activities of elderly people to support independent living [5, 7, 8]. Most of the latest progresses to increase independence of the elders at home are linked to the advancement of software architectures, hardware architectures, remote sensing technologies, constant monitoring algorithms, low power consuming network devices, efficient routing protocols and finally the most significant virtualization of the WSN. In order to efficiently control over WSN services and infrastructure, WSN virtualization is catching huge attention from the researchers [9, 10]. WSN virtualization enables multiple applications to run on a single sensor WSN. In case of AAL WSN virtualization can enable all the healthcare applications to run over a single infrastructure. WSN virtualization forms services by combining multiple sensed data. This service data is further forwarded to the sink. Finally, the service data is transmitted to the device that is running application.

WSN virtualization is attained by adding the middleware layer at the sensor node as well as at the gateway. The role of the middleware layer is to efficiently manage WSN services, infrastructure and support plug and play processes. In virtualized environment the service formation and the middleware layer possess overhead challenges of delay, power utilization, additional memory and extra processing. To cope with the challenges of WSN virtualization in AAL the role of middleware layer needs to be enriched and the middleware layer must provide flexible high level interfaces to application designers so that they can program the network and services of AAL efficiently and simultaneously controlling the overheads of the Additional layer.

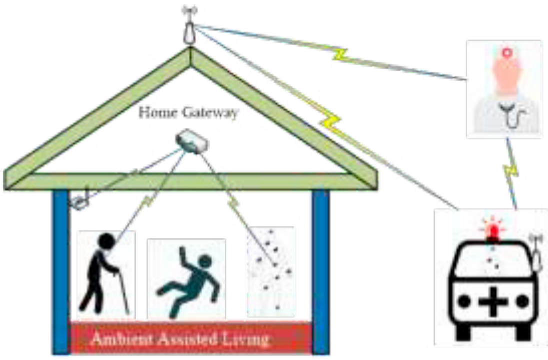

Fig.1 Ambient Assisted Living

Middleware for WSN virtualization stresses on protocols and packet structure and links the higher layer to the lower layers of WSNs. Middleware can efficiently perform the splitting, adding and deleting of the nodes as shown in reference [9]. There have been a number of efforts put forwarded by the researchers to come up with an efficient middleware design. Although, the recent research carried out the paradigm of WSNs virtualization by the reference [11-19] are inspiring. Though, the research challenges regarding effective management, sharing, reducing delay and use of the WSN resources and energy efficiency are still challenging areas of the research. There is a requirement of an efficient middleware design for sensors virtualization that deals with the aggregation of the sensors' data, while decreasing the delay. The efforts have been put forward in this paper and a novel middleware framework is proposed that provides the sensor virtualization and reduces the delay of essential healthcare data for AAL application.

The rest of the research paper is planned as follow: Section II discusses the allied recent research work. Section III presents the offered WSN virtualization middleware layer design and results. Finally, section IV presents the conclusion of the research article.

2. Related Work

In recent past, quite a number of WSN middleware designs have been released. Mainly there are two major classifications of the middleware layer one: middleware for sensor virtualization and Second: middleware for network virtualization. In case of sensor virtualization different services are formed by combining the sensed data from multiple heterogeneous sensors nodes to provide offered services to the application users [20, 21].

Physical sensor node can offer numerous services; Moreover, the offered services depends on the mounted sensors on the sensing node, like: the number of on board sensors, processing power, on board memory [22, 23]. In case of WSN virtualization the application layer is decoupled from the physical sensors. The decoupling of the layers allows the virtual sensors and services to run on top of the physical sensor nodes.

Sensor nodes cannot perform any task if they do not collaborate with each other. The effective cooperation between the sensing devices allows the WSN to achieve higher goals [2]. WSN virtualization deals with different groups of sensors committed to a particular task supporting a particular application [24]. WSN virtualization enables the subset of sensor nodes to accomplish the application tasks, simultaneously other subset of sensor nodes execute the tasks of other different applications [25]. Virtual machine (VM) design method is one of the most effective methods that assist execution of multiplications. Mate middleware [26] takes up the challenge of sensor virtualization. Mate middleware is designed such that the most of the processing is done at the sensor node which is highly appropriate for event managing programming. Melete took a step headfirst in [27]. It included collaboration of sensor nodes to facilitate multiple applications.

VITRO in [11] proposes a service delivering structure for WSN virtualization.A middleware design providing network and sensor virtualization is designed in SenShare middleware [16] and is based on hardware abstraction layer, which separates the node's hardware from the services. The SenShare [16, 28] project from Cambridge University has proposed the middleware running multi applications instead of traditional single application WSN [29]. SenShare middleware takes on the challenge dealing with the services management and executing the tasks of single application on different multiple WSN nodes. The Servilla middleware uses the Tiny operating system, it allows the discovery of services and use local and remote services [17]. Furthermore, Servilla is aimed to provide solution for the heterogeneity of multiple sensing devices, and enables them such that a number of applications can run over multiple heterogeneous nodes. Furthermore, Servilla middleware layer enables network programmer to cultivate a platform that is independent of the applications.

A smart home based middleware designed on the concept of virtualization is presented by the authors in [18]. Moreover, it provides a guide line for multiple infrastructure providers and service providers. The multi-layer middleware architecture is presented in [15, 17] which provides network virtualization, where more powerful sensor nodes are receive more load. PRESTO [30] proposes a middleware architecture based on a predictive proxy centered technique. Furthermore, it provides storage architecture for a large and small WSNs. PRESTO middleware is inspired by the initial work of the COUGAR middleware [31] and the TinyDB middleware [32].

The Hourglass middleware [33] uses SOA, supporting a number of applications. The MiLAN [34] middleware is application-driven Middleware. MiLAN is based on a reliability matrix that groups together several sensors to provide effective and reliable drug support. However, involving more sensors increases reliability, but this approach involves more cost and power consumption because more sensor transmissions take place. Although the recent literature provides many state-of-the-art designs, there is still need of a middleware design of WSN virtualization, where the WSN sink supports most of the network's processing and storage with reduced delay.

3. Proposed Middleware Design

Proposed middleware design is based on the modular and components design approach. Proposed middleware design resides at the gateway and at the sensor node. Middleware design addresses the challenge of efficiently managing WSN and supporting a number of applications while keeping an eye on delay. The goal of proposed middleware design is to provide sensor virtualization and minimize the delay.

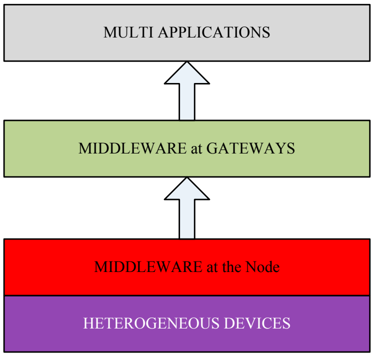

Fig. 2 Proposed Layering Architecture of WSN Virtualization

In Fig. 2 proposed layered view of the middleware deign is shown. The Proposed architecture consists of four layers, from bottom the first layer consists of multiple heterogeneous sensor nodes. Each node consists of multiple onboard sensors. The second layer consists of middleware at the node. Furthermore, third layer consists of middleware at the gateway and the final layer consists of the multiple applications that are running at the user end.

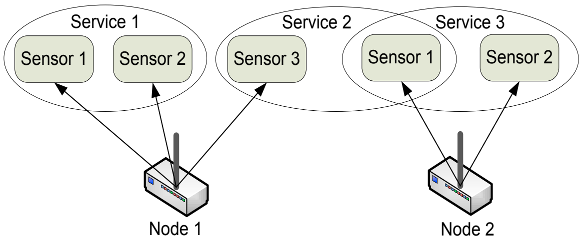

Fig. 3 Proposed Service Formation Structure

Different services can be formed using multiple onboard sensors. Furthermore, these sensors can be on the same node or different nodes as shown in the Fig 3. In addition, only a single sensor node is transformed into a number of logical instances used in multiple applications. In the proposed middleware services help several heterogeneous applications according to QoS needs. Proposed middleware framework is designed to adapt to the changes of environment, depending on the different applications and needs of the different service are formed as shown in the Fig 3. The main requirement in this case is to minimize delay. To overcome the delay, the SOA model is applied. The middleware components required to handle the needs of the applications are only called and executed, instead of the monolithic programming model which runs the entire program. To further reduce the delay and speed up the multithreaded service processing technique this process is applied multiple applications. Although, the multi-threading technique consumes more memory, on the other hand it reduces the processing delay. The processing of each application is isolated from the other application and health care service packets are processed with high priority. The effect of proposed middleware design can be seen by linking the delay of the traditional approach using a single application and using the virtualization approach without using the multi-threading and the proposed middleware technique using the multithreaded technique in the virtualized environment. In the equations t_p is time required to execute a single application and t_s is the reading time of the sensor data. In the traditional approach for$n$applications, the total time taken is indicated by equation 1.

$$T = n(t_p + t_s) \quad (1)$$

Using the middleware technique proposed in [16]the total time it takes for$n$applications is reduced. Equation 2 shows the technique used in [16].

$$T = nt_p + t_s \quad (2)$$

The proposed middleware design uses multi-threading which takes the symbol x as the number of threads in the application and further reduces processing. The mathematical technique of the proposed middleware is shown in equation 3.

$$T = n\frac{t_p}{x} + t_s \quad (3)$$

In the proposed framework, middleware using multi-threading technique reduces the delay.

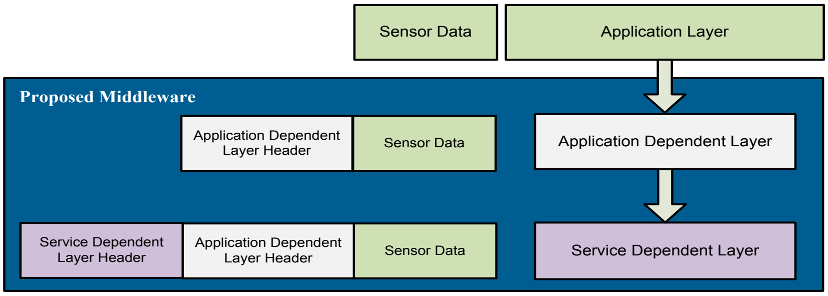

Fig. 4 Middleware Lyer Header Positions in WSN Packet

To further reduce the delay the proposed middleware is divided into application dependent layer and service dependent layer. Each layer adds its header to the application packet as shown in the Fig 4. Service dependent layer is in charge for providing QoS to the high priority applications. Service dependent layer is further composed of three managers namely Service Handling manager, QoS manager and Service Scheduling manager. Moreover, each manager is composed of further different components as shown in the Fig. 5.

3.1. QoS Manager

QoS manager receives the service frame from application dependent layer, after receiving the service frame the Service Identification Component identifies the service type and Priority Identification Component assigns the service frame with the priority level. There are three priority levels set by the Priority Identification Component namely high, medium and low. The service frame dealing with the AAL is always given the high priority, thus reducing the overall delay from the patient to the hospital.

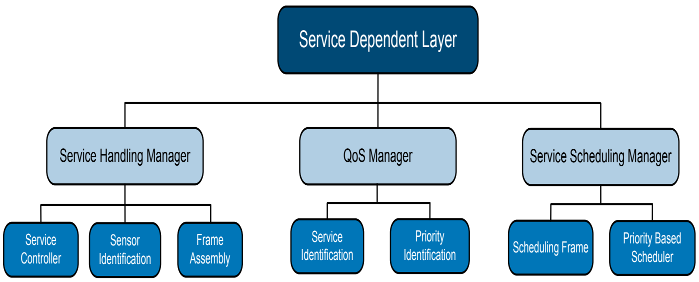

Fig. 5 AAL Middleware Managers and Components

3.2. Service Scheduling Manager

Service scheduling manager deals with the scheduling of the service frame based on priority level assigned by the QoS manager. Service scheduling manager consists of priority based scheduler that is implemented by the priority queue and all the frames have to pass through the priority queue, before the execution. Due to the assigned priority the waiting time for the high priority frame is set to minimum, and as soon as the high priority frame arrives it is processed at earliest. Moreover, the medium priority packets will have to wait for the high priority packets to be processed first. Finally, the low priority frame will have to wait for the medium and high priority frames to be served, before their turn. The arrival rates for different priority frame is represented by, λ_H for the high priority frame, λ_M for the medium priority frame and finally λ_L for the low priority frame. The total sum of arrival rates is given by the equation 4.

$$\lambda_{\mathrm{Sum}} = \lambda_{\mathrm{H}} + \lambda_M + \lambda_{\mathrm{L}} \tag{4}$$

Using a mathematical formulation of M/G/1 that is the closest to the real world, the waiting time of the high priority packet $\mathrm{W_H}$ is given by the equation 5, where $\mathrm{N_H}$ is the number of high priority packets in the queue and μ is the processing rate.

$$\mathrm{W_H} = \frac{\mathrm{N_H}}{\mu} \tag{5}$$

Similarly, the waiting time for the medium priority packets $\mathrm{W_M}$ is given by equation 6.

$$\mathrm{W_M} = \frac{\lambda_H \mathrm{R} + \mathrm{N_H} + \mathrm{N_M}}{\mu - \lambda_{\mathrm{H}}} \tag{6}$$

Finally, the waiting time for the low priority packets $\mathrm{W_L}$ is given by equation 7.

$$\mathrm{W_L} = \frac{\lambda_M \mathrm{R} + \mathrm{N_H} + \mathrm{N_M} + \mathrm{N_L} + \lambda_{\mathrm{H}} \mathrm{W_M}}{\mu - \lambda_M} \tag{7}$$

The analysis of the mathematical model and the real testbed results show that the high priority packets have the least delay and the proposed middleware gives high priority to the AAL service data so the delay is reduced remarkably. The graph in Fig 6 shows measured and calculated values of the waiting time of the packets. The graph clearly shows that the measured values from the testbed are very close to the calculated values. Moreover, Fig. 6 shows that high priority packets have the least delay compared to the medium and low priority packets. The mean waiting time of the high priority frame is 0.88ms, medium priority frame is 44.14ms, low priority frame is 75.3ms and without any priority is 27.46ms.The waiting for the high priority frame is reduced by 97.08%. However, waiting time for the medium priority frame is increased by 58.82% and waiting time for the low priority frame increased by 174.9% when compared to FIFO queue.

3.3. Ambient Assisted Living Testbed

All the experimentation was carried out on the real time AAL testbed. Hardware platform consists of distinct types of nodes having multiple sensors mounted on top of them and a powerful node that works as a gateway. Multiple strong and weak sensor nodes are the part of AAL testbed. It includes sensor nodes using IEEE 802.15.4 and WiFi radios. The experiment on AAL testbed was carried out using TelG sensor nodes and Raspberry Pi node. All the sensor nodes were put in stationary positions.

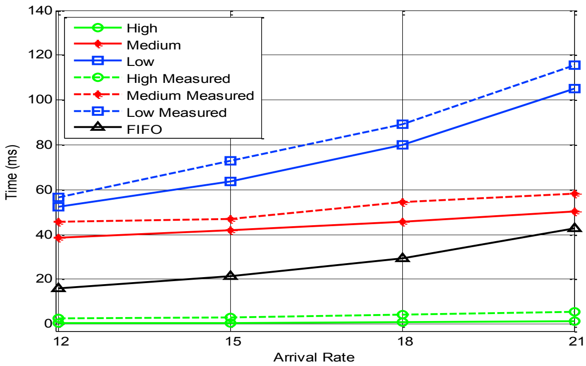

Fig. 6 Priority Based Waiting Time of Packets (Measured and Calculated)

UDOO is used as the gateway device as it offers high processing speed and large memory, which allows middleware computations at the gateway to be carried out in an efficient manner. The default configurations and driver settings of the UDOO's Embedded Linux kernel do not support a wide range of network hardware. For the implementation of the proposed middleware, UDOO's kernel has been re-configured to support multiple network devices. Fig. 7 shows different testbed components, these components include sensor nodes mounted with multiple sensors and the gateway attached with multiple network dongles. The proposed middleware was successfully implemented and run on the AAL testbed.

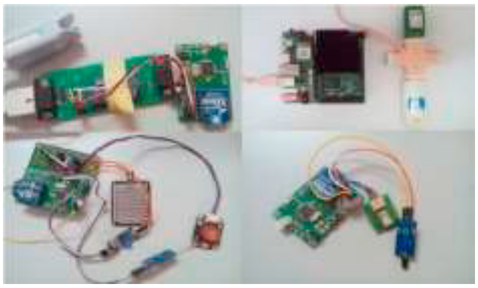

Fig. 7 AAL Testbed Nodes and Gateway

4. Conclusion

In the paper the proposed middleware design for AAL is based on the concept of the sensor virtualization that allows many autonomous applications to operate in parallel. The proposed middleware resides at the sensor node and the gateway. Node's middleware assigns a priority level to the service packet, and implements a priority queue at the packet processing and transmission stage. Similarly, middleware at the gateway processes the packet according the priority level assigned by the node and executes the packet using multi-threading technique. The proposed middleware design improves overall WSN processing time by simultaneously running multiple different service daemons. The middleware processing on the node and the gateway speed up the processing of the AAL packet and reduce the delay significantly. Reduction of WSN latency is achieved by the middleware which, in turn, allows real-time applications like AAL to run efficiently. The evaluation of the middleware has been done in AAL test bed environment.

Acknowledgements

The authors would like to thank Federal Urdu University of Arts Science and Technology (FUUAST), Universiti Teknologi Malaysia (UTM) and HEC Pakistan for their support.

References

[1] D. Piromalis and K. Arvanitis, "Sensotube: A scalable hardware design architecture for wireless sensors and actuators networks nodes in the agricultural domain," Sensors, vol. 16, p. 1227, 2016.

[2] Z. Khalid, N. Fisal, and M. Rozaini, "A Survey of Middleware for Sensor and Network Virtualization," Sensors, vol. 14, pp. 24046-24097, 2014.

[3] T. P. Lambrou and C. G. Panayiotou, "A testbed for coverage control using mixed wireless sensor networks," Journal of Network and Computer Applications, vol. 35, pp. 527-537, 2012.

[4] I. Bojic, G. Jezic, D. Katusic, S. Desic, M. Kusek, and D. Huljenic, "Communication in machine-to-machine environments," in Proceedings of the Fifth Balkan Conference in Informatics, 2012, pp. 283-286.

[5] P. Rashidi and A. Mihailidis, "A Survey on Ambient Assisted Living Tools for Older Adults," IEEE Journal of Biomedical and Health Informatics, vol. 17, p. 11, 2013.

[6] S. Fleck and W. Straßer, "Smart camera based monitoring system and its application to assisted living," Proceedings of the IEEE, vol. 96, pp.

1698-1714, 2008.
[7] R. I. Sokullu and A. Balcı, "M2M Communications and Their Role in AAL," in Ambient Assisted Living and Enhanced Living Environments, ed: Elsevier, 2017, pp. 441-487.
[8] Z. Khalid, N. Fisal, H. Safdar, W. Maqbool, S. Zubair, and A. Khan, "M2M Communication in Virtual Sensor Network for SHAAL," Jurnal Teknologi, vol. 65, 2013.
[9] Y. K. Seung Hwan Shi, R. Rim Haw, "Intelligent M2M network using healthcare sensors," Network Operations and Management Symposium (APNOMS), 2012 14th Asia-Pacific, pp. 1 - 4, 2011.
[10] Z. Khalid, N. Fisal, H. Safdar, R. Ullah, and W. Maqbool, "System Design in Sensor Network Virtualization for SHAAL."
[11] L. Sarakis, T. Zahariadis, H.-C. Leligou, and M. Dohler, "A framework for service provisioning in virtual sensor networks," EURASIP Journal on Wireless Communications and Networking, vol. 2012, pp. 1-19, 2012.
[12]D. D. Vergados, "Service personalization for assistive living in a mobile ambient healthcare-networked environment," Personal and Ubiquitous Computing, vol. 14, pp. 575-590, 2010.
[13] Q. Wang, W. Shin, X. Liu, Z. Zeng, C. Oh, B. K. AlShebli, et al., "I-Living: An open system architecture for assisted living," in Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on, 2006, pp. 4268-4275.
[14] E.-N. Huh, "An Efficient Model for Smart Home by the Virtualization of Wireless Sensor Network," International Journal of Distributed Sensor Networks, vol. 2013, 2013.
[15] I. Khan, F. Belqasmi, R. Glitho, and N. Crespi, "A multi-layer architecture for wireless sensor network virtualization," in Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP, 2013, pp. 1-4.
[16] I. Leontiadis, C. Efstratiou, C. Mascolo, and J. Crowcroft, "SenShare: transforming sensor networks into multi-application sensing infrastructures," in Wireless Sensor Networks, ed: Springer, 2012, pp. 65-81.
[17] C.-L. Fok, G.-C. Roman, and C. Lu, "Enhanced coordination in sensor networks through flexible service provisioning," in Coordination Models and Languages, 2009, pp. 66-85.
[18] M. M. Islam, J. H. Lee, and E.-N. Huh, "An Efficient Model for Smart Home by the Virtualization of Wireless Sensor Network," International Journal of Distributed Sensor Networks, vol. 2013, p. 10, 2013.
[19] A. Pantelopoulos and N. G. Bourbakis, "A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 40, pp. 1-12, 2010.
[20] S. Alam, M. M. Chowdhury, and J. Noll, "Virtualizing sensor for the enablement of semantic-aware internet of things ecosystem," Guest Editors, p. 41, 2010.
[21] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: A survey," IEEE Communications Surveys & Tutorials, vol. 18, pp. 553-576, 2016.
[22] Z. Khalid, N. Fisal, M. Rozaini, H. Safdar, R. Ullah, and W. Maqbool, "Middleware framework for network virtualization in SHAAL," in Computer Applications and Industrial Electronics (ISCAIE), 2014 IEEE Symposium on, 2014, pp. 175-179.
[23] Z. Khalid, N. Fisal, S. Zubair, R. Ullah, H. Safdar, W. Maqbool, et al., "Multi-thread based middleware for sensor network virtualization," in Information Technology: Towards New Smart World (NSITNSW), 2015 5th National Symposium on, 2015, pp. 1-6.
[24] S. Kabadayi, A. Pridgen, and C. Julien, "Virtual sensors: Abstracting data from physical sensors," in Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks, 2006, pp. 587-592.
[25] A. P. Jayasumana, Q. Han, and T. H. Illangasekare, "Virtual sensor networks-A resource efficient approach for concurrent applications," in Information Technology, 2007. ITNG'07. Fourth International Conference on, 2007, pp. 111-115.
[26] P. Levis and D. Culler, "Maté: A tiny virtual machine for sensor networks," in ACM Sigplan Notices, 2002, pp. 85-95.
[27] Y. Yu, L. J. Rittle, V. Bhandari, and J. B. LeBrun, "Supporting concurrent applications in wireless sensor networks," presented at the Proceedings of the 4th international conference on Embedded networked sensor systems, Boulder, Colorado, USA, 2006.
[28] C. Efstratiou, I. Leontiadis, C. Mascolo, and J. Crowcroft, "A shared sensor network infrastructure," in Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, 2010, pp. 367-368.
[29] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," Computer networks, vol. 52, pp. 2292-2330, 2008.
[30] P. Desnoyers, D. Ganesan, , and P. J. Shenoy, "PRESTO: A Predictive Storage Architecture for Sensor Networks," in HotOS, 2005.
[31] Y. Yao and, "The cougar approach to in-network query processing in sensor networks," ACM Sigmod Record, vol. 31, pp. 9-18, 2002.
[32] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: an acquisitional query processing system for sensor networks," ACM Transactions on database systems (TODS), vol. 30, pp. 122-173, 2005.
[33] J. Shneidman, P. Pietzuch, J. Ledlie, M. Roussopoulos, M. Seltzer, and M. Welsh, "Hourglass: An infrastructure for connecting sensor networks and applications," Tech. report, Harvard TR-21-042004.
[34] A. Murphy and W. Heinzelman, "Milan: Middleware linking applications and networks," University of Rochester, Tech. Rep. TR-795, 2002.