Introducing Autonomy in Internet of Things

QAZI MAMOON ASHRAF AND MOHAMED HADI HABAEBI
Department of Electrical and Computer Engineering
International Islamic University Malaysia
Jalan Gombak, Selangor
MALAYSIA
mamoonq@gmail.com; habaebi@iium.edu.my

Abstract: - Internet of Things (IoT) is playing a major role in extending the reach of the existing communication systems to include resource constrained devices. Many exciting research works for IoT have been proposed for management of such devices such that human intervention is minimized. This is a challenge due to the high heterogeneity, high complexity of the devices and the lack of dynamic management schemes.Here, we introduce the paradigm of autonomic computing to be used for such dynamic yet secure management in IoT. The adoption of the autonomy in IoT architecture can prove to be a valuable addition to IoT systems.

Key-Words: -Internet of Things, Autonomy, Management, Wireless Sensor Networks

1 Introduction

The deployment of a large number of sensor devices in various applications has led to the arrival of Internet of Things (IoT). The manual installation and management of these devices becomesimpractical due to the large numbers involved.Specifically, there exists an inefficiency that can be resolved by minimizing user intervention.The manual maintenance of a large number of devices becomes inefficient, and demands the presence of intelligent and dynamic management schemes.

A similar problem has been encountered in traditional client-server paradigms, where autonomic computing has come to the rescue. Autonomic computing has substantially helped minimize user intervention for management of computer systems. In the traditional client-server paradigm, and enterprise computing, an autonomic system is defined as "*an intelligent system, or system of systems where data acquired by sensing or monitoring capability is utilized in an overall autonomic decision-making process.*"[1]. Such a system should be able to vary its configuration dynamically throughout its working duration. The main goal of autonomic computing is the management of computing resources in a manner so as to minimize the user intervention. It is essentially a group of computer systems which are managing and optimizing the functions of other groups of computing systems. All this is done while minimizing manual intervention[2].

Autonomic systems have been defined differently by many researchers. For our purpose, we accept the definition provided in[1]. The reason of selecting this definition is because it can be applied easily for the scope of IoT, and is not limited by specific issues.

Autonomy in IoT can be realized by implementing self-managing systems. Self-management is the property of a system to achieve management and maintenance of its resources intrinsically and internally. Management and maintenance is realized through many levels of decision making. In IoT, the management scope extends to access management, device management as well as service management. Thus, for self-management, decision making in IoT should pertain to this management scope of IoT. As a consequence, autonomic IoT will be achieved after self-management is achieved. Later in this paper, we expand the reasoning behind this and describe the architecture to introduce autonomy into the traditional IoT architecture.We also discuss other scopes of management such as data management, context management as well as trust management.

2 Related Work

Recently, Perera et al.[3][4][5] have worked on similar management problem sets using few middleware approaches for decision making. For decision making, parameters of device context information are gathered to allow heterogeneous devices to be manageable under a common framework. Sensor data is filtered at the application layer where the complete data is stored.The major disadvantage of this method is that the end network

Table I.Feature comparison of autonomic behavior with literature

Method	[10]	[9]	[5]	Autonomic Scheme
Approach	Middleware	SDN	Semantics	System
Component Based Design	Yes	-	Yes	Yes
Support for Constrained, Heterogenous Nodes	Yes	No	Yes	Yes
System Topology Aware	No	Yes	No	Yes
Node Registration	No	Possible	No	Yes
Service Self-Configuration	Possible	No	No	Yes

continues to forward all the data, irrespective of the fact that only few data sets are required. Furthermore, individual nodes can't be requested for specific data.This leads to a situation where all the end nodes are active and adds a constraint to energy conservation by affecting the duty cycle.

Krco et al. [6]designed a system to achieve "*plug-and-play"* functionality via a middle ware implementation. This middleware acts as a dynamic manager which allows the system to quickly add or remove sensors and networks. The end nodes are essentially the sensors and the networks are essentially the gateway along with their set of end nodes. All the data is forwarded to the gateway, which acts as a wireless network sink. In turn, the gateway groups the data and forwards it to the machine management system.

Rajan et al. [7] first realized that an intelligent management scheme is required for device management in IoT. They introduce a scheme and present general requirement for achieving the same, without much attention to technical specifications.

Pujolle G. [8] proposes an IoT architecture following similar requirements and principles. The scope of decision making is limited to selecting the best communication protocol to use. A higher level overview on agent features is provided without enough details on how autonomy is achieved in the protocol. The performance evaluation for IoT architecture, however,was conducted in a telephone switching system instead of a wireless network.

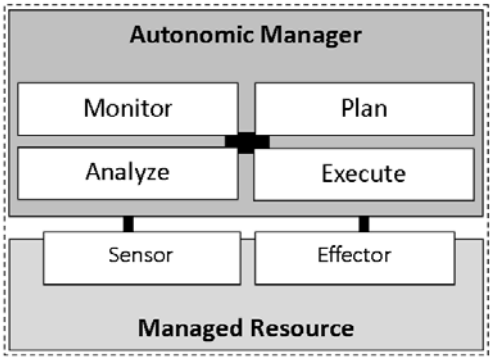

Fig. 1. The autonomic control loop divided between an autonomic manager and a managed resource. Adapted from[11].

Features such as topology control, node registration as well as support for heterogeneous, constrained nodesare compared in Table I for few other research works as well. Few requirements of autonomy are mentioned and the research works are compared against the same. A component based design is required for autonomy as it allows for modularity in the design. The feature of the support for constrained, heterogeneous nodes is quite broad, and eventually decision making for management of such nodes is important. In this regard, our goals and those of[5] are similar. An autonomic system should be system topology aware. This property allows to support both dynamic control as well being environmentally aware. As an example, the SDN based approach [9] offers a high level of such networking capabilities. Autonomy can also be realized in the field of service self-configuration and service management. For the middleware proposed in[10], suitable modifications can allow this approach to support autonomy in service configuration.

3 Autonomic Framework

IBM[11], in the year 2003, first introduced the theory of autonomic computing. A framework was proposed which was aimed to make the management of systems easier. This was done making the management resources in the system less dependable on human input. The framework consists of two functionally important entities, namely, 1) Managed Resource and 2) Autonomic Manager.

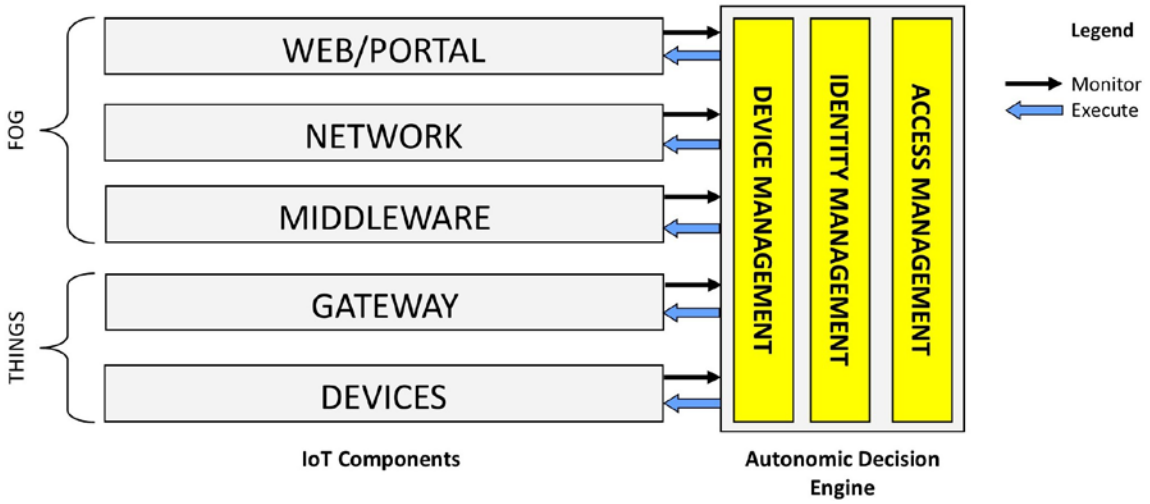

Fig. 2. The autonomic component system interaction for management using an autonomic scheme.

In the autonomic framework, the two entities are separated by the level of complexity and the role in the overall decision making process. The managed resource is the less complex entity whereas the autonomic manager is highly complex and provides the central control and analysis. The managed resource acts as a monitoring interface for the environmental states. Thus, their function can be described primarily as data collectors. In addition to these two entities, an effector interface in the managed resource allows for the manipulation of the environment by the system. Effectors are essentially actuators which act upon command to change the current state in the system and environment.

Autonomy in the managed resource and the managed elements is implemented through the control loop of monitor, analyze, plan and execute. These modules are presented in Fig. 1. The monitor component, as the name suggests, is responsible to collect details from a managed resource element, as well as the environment. The whole purpose is to aggregate, filter, manage, and report the data the autonomic manager. On the other hand, the analyze components allows to play with the monitored data, and extract useful patterns and information. This processing enables the system to learn about the environment and even predict future situations. The plan component provides further mechanisms to guide the action with the help of higher level policies, in order to achieve goals and objectives. These are high level objectives set up by the system designer or the system administrator. Finally, the execute component controls execution of pre-defined plans and interfaces with the managed resource.

4 Autonomy in IoT

Figure 2 shows the relation between the IoT and autonomic decision making. The left side of the figure shows the traditional layers and components of IoT. These are grouped as the *fog* and the *things*. The fog is a concept that computational resources that traditionally used to exist on the cloud are coming further down to the end devices, thereby being termed as *fog*. In this case, the *fog* comprises of the middleware, the network and the virtualization capabilities, as well as the web applications and portals. The *things* group comprise of gateways and end devices.

Implementing the control loop of autonomy means to assign autonomic managers and managed resources to these components. Interestingly, the flexibility of IoT allows to assign the autonomic paradigm components to any IoT components, as long as hierarchy is maintained. Thus, each component above can be the autonomic manager of the components below, either individually or as a group. Autonomy will assist in decision making for functions of device management, access management, as well as identity management. These problems are manifest throughout each layer of IoT. The reasoning is that there is need to manage end devices, gateways, multiple middlewares as well as remote servers. Each set of components can be governed and made more efficient using the autonomic framework. Constant monitoring allows real time decision making and execution of tasks.

For autonomy to be introduced into IoT, it is important to copy the autonomic framework and

ISBN: 978-1-61804-297-2

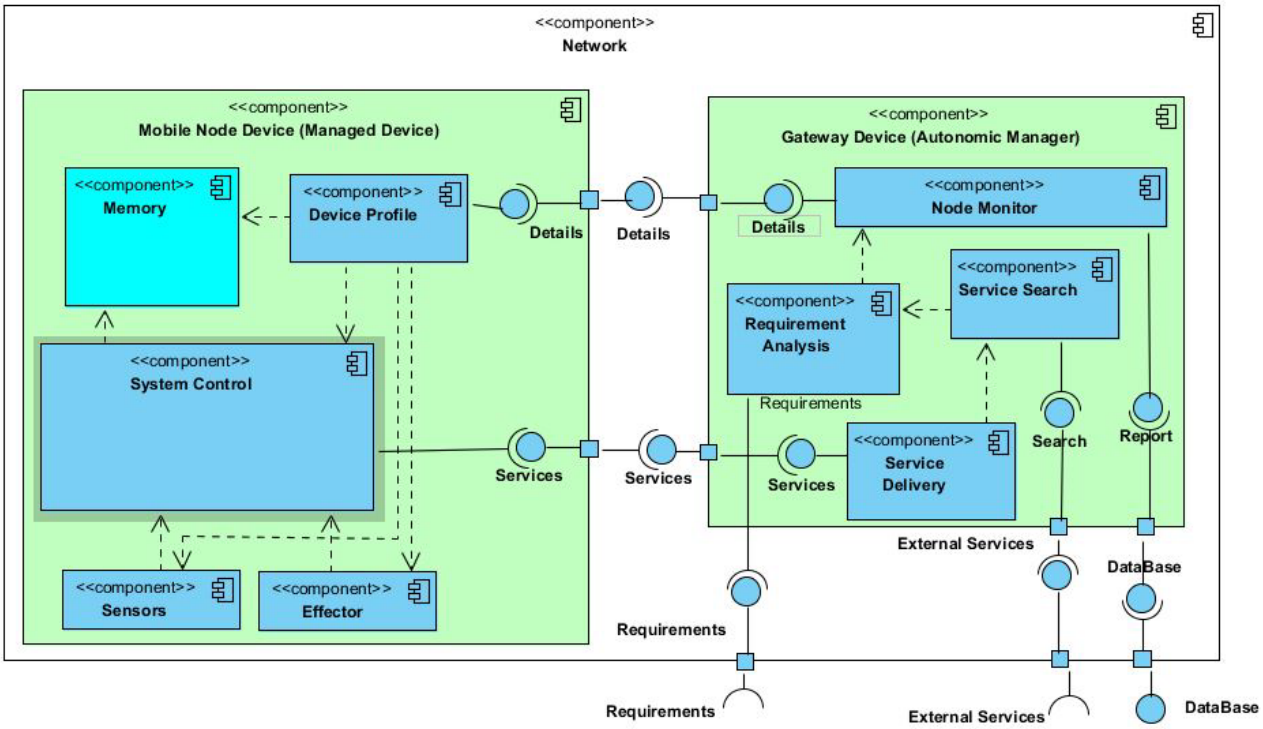

Fig. 3.The overall component architecture and their dependencies. Adapted from [2]

apply into the structure of IoT as well. We take the example of a gateway and devices pair. Here we assign the gateway to be an autonomic manager and the devices to be the managed resources. The concept of managed resource has been previously applied [2] and its role is represented by the mobile node device (MND). The role of autonomic manager on the other hand is fulfilled by the gateway device (GWD). Therefore, we denote a MND as an element that contains sensors and effectors for sensing and interacting with the system environment. We also denote GWD to act as the autonomic manager.

4.1 Mobile Node Device (MND):

To facilitate an autonomic system, it is necessary to have the following components present in the end device: These include a memory component, a device profile component and a system control component.

1. Memory: Memory and its management is an essential part for implementing autonomy as all the services that are to be received will be stored in the memory. Services will be received using the configuration information for being stored in the memory.

2. Device Profile: This component will keep track of the current status of the MND. Information about the memory along with few other parameters will be sent to the GWD so that it can make suitable decision for what services to offer or deploy. This contributes directly to the monitoring stage in the control loop.

3. System Control: This component is for the overall control and management in the MND and can be an operating system or a process inside an operating system. For having the self-configuring feature, it requires an interface of services information from the GWD. Such a module contributes directly to the execution stage in the autonomic control loop.

4.2 Gateway Device (GWD)

We denote a GWD as an element that contains features to act as an autonomic manager to achieve the functionalities of plan, analyze, monitor and execute. To facilitate an autonomic system, it is necessary to have the following components present in the autonomic manager. The components proposed inside the GWD are a node monitor component, a requirement analysis component, a service search component, and a service delivery component.

1. Node Monitor: This component in a GWD is responsible to receive and collect the details from the MND. It accumulates the data locally, filters, and forwards all the data through an interface to an external data base component.

2. Requirements Analysis: This component in a GWD is responsible to obtain the requirements

from an external source or by a local decision. The GWD may set up its own mandated goals and levels or services or may be able to receive the requirements from an external component. This component is also responsible to understand the functionality of the MND that is currently under perspective leading to arrive at a decision. This relates to the analyze module in the autonomic control loop.

3. Service Search: Based on the decision from the 'Requirements Analysis' component, this component will look for suitable services.

4. Service Delivery: Once the service required has been finalized, the configuration elements are delivered to the MND. This also relates to the execution stage in the autonomic control loop.

The final proposed system architecture is presented as shown in Figure 3.

5 Guidelines for Secure Autonomy

Security not only encompasses privacy, but also availability and integrity. For network security and rigidness, an autonomic IoT network would need some routing capability, as a backup method to route the data [12]. So instead of just behaving as an add-on to the existing infrastructure of the Internet, guidelines and policies are required to make it self-sufficient and practical. We present some scenarios where autonomic decision making can contribute in terms of security in IoT.

5.1 Storage Management

The autonomic system should be able to decide dynamically about the amount of data to be stored locally and remotely based on the external conditions. This decision scope will be related to storage management. Dynamically setting an optimum or minimal storage use and encrypting the storage for confidentiality will the foremost priority. Perhaps, the ability to re-generate lost data may not be achieved in the present state of technology, but detection of such an event can allow further actions to be taken so as to minimize the damage.

5.2 Logging Data Paths

It is essential for an autonomic system in IoT to generate logs keeping tracking of the major as well as minor events in the system. This will allow the exposure of the path of data and any alteration or fault can be traced back to its root cause.Decision making in this case is concerned on how to treat the logged data, as well as how often to log the data. There may also be different levels of logging, locally done in the managed resource as well as done in the remote server for the whole system.

5.3 Integrity of Device Firmware

After going through pattern of data communication, an attacker may be able to create another firmware to mimic the capabilities of genuine nodes. It will be highly advantageous if the autonomic system can make sure that all devices will run only authorized software. Thus, there should be additional headers and control data that will need to be generated to monitor the system.

5.4Minimal Functionality

No system is fool-proof, and failures and faults are always to be expected. In autonomic IoT, once a failure or attack occurs, self-healing functionality should be enabled such that systems should at least be able to deliver the lowest level of functionality.

5.5 Scalability

In the event of introducing and including extra resources in an IoT network, the expansion should occur smoothly. This is also known as the scalability issue in IoT.Autonomy can assist in the scalability issue by deciding on duty cycling methods, where part of the network can be switched off without losing functionality. Autonomic decision making, in this case, can assist in prolonging the lifetime of the network without loss of availability and functionality. As an example how scalability and availability are inter-related, we present the work done in [13]. Here, a large number of nodes attempt to enter and register to a network simultaneously. This result in repeated collisions, and thereby loss in the network availability. In this case, the system can automatically decide on contention parameters using an autonomic engine.

5.6 Non-linkability

As a part of autonomic data management, non-linkability refers to the separation of the data belonging to the same user or same device, such that the data may not allow a third party to establish a profile of the owner.

A single user may also own a multitude of devices. In this case, an intelligent autonomic system should be able to dynamically add noise to the data, and then be able to filter it out as well. This will prevent any attacker from searching for patterns and reverse engineering any sniffed data. The major disadvantage that arises with the implementation of such a scheme is the increase in bandwidth utilization. Here, the autonomic system has to

decide on the addition of data noise, as well as the channel frequency, such as in cognitive aware systems.

5.7 Context Privacy

As a part of autonomic access management, the access context information of the end devices should be kept secret. It is important for the autonomic system to ensure that personal as well as device data is protected. In [14], there exist varying levels of profiles for such context privacy. There may be few data sets which can be accessed by a doctor without a patient's permission, and there may be other sets which the doctors always have access to. Similarly, other stakeholders in a medical IoT system will have varying levels of access to a patient's records.

5.8Preserving Anonymity

The purpose of preserving anonymity for IoT end nodes is that the identity of a node be hidden for any third parties. It is important for the autonomic system to decide in a way that identity in not compromised. However, a purely anonymous communication is not possible because of shortcomings of existing communication protocols, such as the need for authentication. Anonymity and authentication are opposite goals.

5.10 Trust Management

Trust management will be a future problem in the autonomic decision making process. This justification for this because a large scale adoption of IoT is proportional to the security offered by IoT services. Trust is one important factor which helps customer acceptance as well as reduce the element of risk.

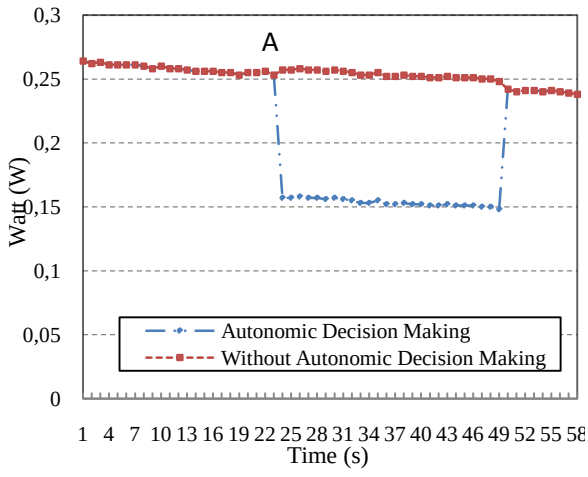

Fig. 4. Test-bed results for the change in battery sensor measurements over time. The difference is shown with and without the autonomy in decision making whether it is required to transmit data or not[2].

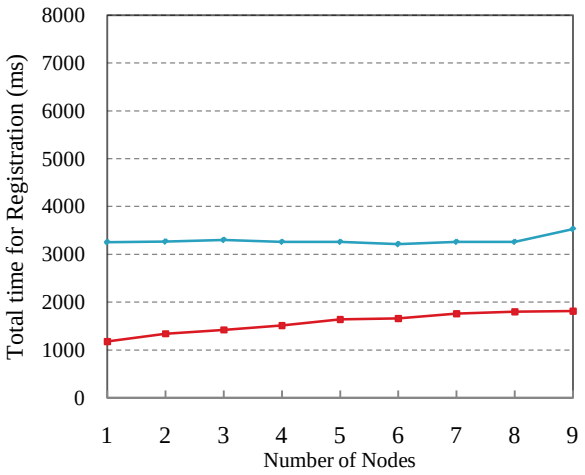

Fig. 5. Time for registration for a number of nodes for autonomic protocol.Simulation results for the registration of a number of IoT nodes with a gateway. The difference is shown with and without the autonomy in decision making whether it is required to transmit data or not. The line at the bottom shows results after autonomic decision making[2].

5.11 Environmental States

An important factor in introducing autonomy in IoT, is that constant monitoring of the environment and functional states is required. A security requirement for such data also arises. The control data to monitor the functional states of any system also needs to be authenticated and protected from manipulation. This can be relevant for detecting device faults, detecting configuration changes, as well as for collecting performance data[12].

6 Preliminary Results

Figure 4 shows the power consumption difference in an actual Zolertia Z1 node before and after having done self-configuration using the autonomic control loop. The configuration was done automatically to enable transmission of sensor readings thereby affecting the duty cycle. The battery level was measured by its internal battery sensor and processed using an available Contiki implementation [15]. The readings were then transmitted to the gateway for storage and display. The autonomic decision making took place at point A, which resulted in a difference in processing. This eventually affected the energy level in the battery.

A decision making scheme isalso simulated using Contiki-based Cooja simulator for emulated Z1 nodes in [2]. The nodes are divided in two sets with each set having different values for the timing

ISBN: 978-1-61804-297-2

variables. The total time taken to register a varying number of nodes (1-9) is found out as shown in Figure 5 [2]. The decision is based on constant monitoring of the environment, and then performing analysis to come up with a decision.

7 Conclusion

Autonomy in IoT is still in infant stages, however researchers have started to recognize the importance of minimizing user intervention. The introduction of autonomic theory in IoT to achieve dynamic management of resource constrained devices by minimizing user intervention is one such solution. At the same time, autonomy permits innovative use of various security schemes. It is only a matter of time until theadoption of the autonomy in IoT transforms functionality, management and energy efficiency in IoT systems.

Acknowledgement:

This research was supported by the Ministry of Higher Education Malaysia (MOHE) under Exploratory Research Grant Scheme (ERGS) number ERGS13-023-0055.

References:

[1] AAD Knowledge Transfer Network, "Autonomous Systems: Opportunities and Challeneges for the UK," 2012. [Online]. Available: https://connect.innovateuk.org/c/document_library/get_file?folderId=278657&name=DLFE-91023.pdf. [Accessed 6 June 2013].

[2] Q. Ashraf, M. Habaebi, G. Sinniah, M. Ahmed, S. Khan and S. Hameed, "Autonomic protocol and architecture for devices in Internet of Things," in *IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, Kuala Lumpur, 2014.

[3] C. Perera, P. Jayaraman, A. Zaslavsky, P. Christen and D. Georgakopoulus, "Context-aware Dynamic Discovery and Configuration of 'Things' in Smart Environments," in *Big Data and Internet of Things: A Roadmap for Smart Environments, Studies in Computational Intelligence*, Springer Berlin Heidelberg, 2014.

[4] C. Perera, P. Jayaraman, A. Zaslavsky, P. Christen and D. Georgakopoulus, "MOSDEN: An Internet of Things Middleware for Resource Constrained Mobile Devices," in *47th Hawaii International Conference on System Sciences (HICSS)*, Kona, 2014.

[5] C. Perera, A. Zaslavsky, P. Christen, M. Compton and D. Georgakopoulos, "Context-Aware Sensor Search, Selection and Ranking Model for Internet of Things Middleware," in *IEEE 14th International Conference on Mobile Data Management (MDM)*, Milan, 2013.

[6] S. Krco, V. Tsiatsis, K. Matusikova, M. Johansson, I. Cubic and R. Glitho, "Mobile Network Supported Wireless Sensor Network Services," in *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS 2007)*, Pisa, 2007.

[7] M. Rajan, P. Balamuralidhar, K. Chethan and M. Swarnahpriyaah, "A Self-Reconfigurable Sensor Network Management System for Internet of Things Paradigm," in *2011 International Conference on Devices and Communications (ICDeCom)*, Mesra, 2011.

[8] G. Pujolle, "An Autonomic-oriented Architecture for the Internet of Things," in *IEEE John Vincent Atanasoff International Symposium on Modern Computing (JVA '06.)*, Sofia, 2006.

[9] W.-Y. Huang, J.-W. Hu, S.-C. Lin, T.-L. Liu, P.-W. Tsai, C.-S. Yang, F. I. Yeh, J. H. Chen and J. Mambretti, "Design and Implementation of an Automatic Network Topology Discovery System for the Future Internet Across Different Domains," in *26th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Fukuoka, 2012.

[10] M. S. Familiar, J. F. Martínez, I. Corredor and C. García-Rubio, "Building service-oriented Smart Infrastructures over Wireless Ad Hoc Sensor Networks: A middleware perspective," *Computer Networks*, vol. 56, no. 4, pp. 1303-1328, 2012.

[11] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41-50, 2003.

[12] Q. M. Ashraf and M. H. Habaebi, "Autonomic schemes for threat mitigation in Internet of Things," *Elsevier Journal of Network and Computer Applications*, vol. 49, no. 1, pp. 112-127, 2015.

[13] Q. M. Ashraf, M. H. Habaebi, G. R. Sinniah and J. Chebil, "Broadcast based registration technique for heterogenous nodes in the IoT," in *International Conference on Control, Engineering, and Information Technology (CEIT 2014)*, Sousse, 2014.

[14] Q. M. Ashraf, M. H. Habaebi and J. Chebil, "SIHAT: simplifying interfaces in health-nets for achieving telemetry," in *Handbook on the emerging trends in scientific research*, Kuala Lumpur, Pak Publishing Group, 2014, pp. 207-217.

[15] C. Alberto Boano, April 2010. [Online]. Available: https://groups.google.com/forum/#!topic/osdeve_mirror_rtos_contiki-developers/jQ3eoMLN02c.

Qazi Mamoon Ashraf is working towards obtaining his Ph.D. in Computer Engineering from the Department of Electrical and Computer Engineering, Universiti Islam Antarabangsa Malaysia. He is also working at Digital Communications Lab, Telekom Research and Development and involved in IoT research in Malaysia. He was also a Research Assistant with Wireless Communication Division in MIMOS, Malaysia. His research interests include Internet of Things, autonomic computing, ubiquitous networks and secure M2M communication.

Mohamed Hadi Habaebi is an Associate Professor and the Post Graduate Academic Advisor at the Department of Electrical and Computer Engineering of University Antarabangsa Malaysia. He received his MS in Electrical Engineering from Universiti Teknologi Malaysia, and his PhD in Computer and Communication System Engineering from Universiti Putra Malaysia. His research interests include M2M communication protocols, wireless sensor and actuator networks, and network performance optimisation.