# Mitigating Attacks in the Internet of Things with a Self-protecting Architecture

Ruan de A. C. Mello, Admilson de R. L. Ribeiro, Fernando M. de Almeida, Edward D. Moreno
Department of Computing
Federal University Sergipe UFS
São Cristóvão, Brazil
E-mails: ruanmello@gmail.com, admilson@ufs.br, fernando.m.al.91@gmail.com, edwdavid@gmail.com

***Abstract*—Internet of Things (IoT) applications run in environments that have resource constraints and are unsecured. Due to the nature of their environment, IoT systems should be able to reason autonomously and take self-protecting decisions. Currently, an adequate architecture to incorporate self-protection in the IoT is not available. Thus, we design a new self-protecting architecture based on the MAPE-K (Monitor, Analyze, Plan, Execute and Knowledge) autonomic control loop that will run at the application layer so that developers can add several security services. In this paper, we address the impact caused by attacks (SinkHole, Selective Forward, Black Hole and Flooding) in relation to power consumption and interference in the operation of the network created from the Routing Protocol for Low Power and Lossy Networks (RPL) routing protocol.**

***Keywords–Autonomic Systems; Self-protecting; IoT; MAPE-K loop; AIS.***

## I. INTRODUCTION

Recently, the integration of embedded systems, wireless networks and the Internet led to a new application type, namely Internet of Things (IoT) applications. A particular case of IoT applications is the participatory sensing application where people that live in communities and are dependent on each other for daily activities exchange information to reach their objectives [1]. Recommendations for a good restaurant, car mechanic, movie, phone plan and so on were and still are some things where community knowledge helps us in determining our actions.

IoT applications will have a great impact on people’s life, but currently only a small number of such applications is available to our society. As the things are nodes of a network, individuals can control, locate, and monitor everyday objects remotely. For example, the use of wireless sensor technologies allows monitoring the health of people in real time, enabling brief diagnostics. The vital parameters of individuals such as blood pressure, temperature, and so on, are measured through sensor nodes that stay on the bodies of patients that continue to do their daily activities [2]. Many benefits can be provided by the IoT technologies in the health-care domain.

However, IoT applications run in environments that have resource constraints and are unsecured. The resources constraint of the IoT devices can lead to security breaches. For example, an attacker can try to maintain the IoT devices in operation all the time, with the intention to consume all their battery energy. This attack is a type of Denial of Service (DoS) attack and can have a great impact on the application availability without the possibility of control by users. Therefore, due to that environment, the security issue must be treated autonomously. That is, the self-protection property must be incorporated in the IoT systems [3].

However, the majority of security mechanisms in IoT is composed of protocols and algorithms that run at the physical layer or link layer of the protocol stack of the software system [4]. These mechanisms are adequate to protect against the problems relating to the confidentiality and the integrity, but in some cases they fail on considering the availability.

Considering the availability of applications, self-protection is the essential property that allows network nodes to communicate and react to attacks of hackers according to security policies defined by users [5]. Thus, IoT systems should be able to reason autonomously and make self-protecting decisions.

Therefore, in this context, we propose a self-protecting architecture for the Internet of Things based on the MAPE-K control loop [5] and the danger theory of the Artificial Intelligent System (AIS) [6]. To show the use of the architecture, we implement the execution phase describing the main attacks in the IoT and their impacts in relation to power consumption and interference in the operation of the network.

The remainder of this paper is organized into nine more sections. Section II presents the limitations of related works and highlights our contribution. Section III presents the autonomic loop MAPE-K. Section IV presents the Routing Protocol for Low Power and Lossy Networks. Section V gives a brief overview of the main attacks that occur in the IoT environment. Section VI outlines our architecture, considering the MAPE-K control loop and its phases are described. Section VII discusses how we implement the execution phase of our architecture. Section VIII presents the results obtained so far. Section IX concludes the paper and presents future works.

## II. RELATED WORK

In the literature, there are several papers about computing security based on the AISs and autonomic computing. Kephart et al. [6] and White et al. [7] designed the first AISs in response to the first virus epidemics, when it was found out that the spreading of cure had to be faster that the contamination by viruses. After, SweetBair [8] used a more sophisticated technique to capture suspecting traffic and generate signatures of worms. As a variant of this pattern, Swimmer [9] and also Rawat and Saxena [10] presented an approach based on danger theory for attack detection in autonomic networks.

SVELTE [11], as the authors claim, is the first Intrusion Detector System (IDS) for the IoT. The work presented has a huge contribution to design an IDS with the characteristics of a network for IoT, considering the technologies used in the communications stack, such as IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) and RPL routing protocol. However, its approach does not have autonomic characteristics for self protect the network from further attacks, only the determined attack on the network project level. Like the SVELT, the CAD [12] not only detects the attack, but also tries to mitigate the damage caused by the attacker. The CAD is directed to the wireless mesh network (WMN) and can differentiate between losses occurring in the normal events of a legitimate attack of the type Selective Forward.

The main limitation is that they are not well suited for the IoT environment. They only present some particular solutions that address a set of specific problems different of the IoT environment. Besides, they do not consider the possibility of development of new self-protecting services. Therefore, programmers are unable to decide which policies are more appropriate for their applications, considering still that the resolved policies of low-level protocols sometimes are not the most appropriate for all applications in the IoT.

Therefore, our solution advances previous solutions providing a new self-protecting architecture for the IoT and also the possibility to extend such architecture with new security services.

## III. MAPE-K AUTONOMIC CONTROL LOOP

In March 2001, Paul Horn presented for the first time the MAPE-K Autonomic Control Loop at an IBM event. The MAPE-K Loop was presented as a reference model. Composed of five modules that can be seen in Figure 1, the MAPEK-K Loop is intended to distribute the tasks of each element of the autonomic computing [13]. The modules that build the MAPE-K Loop are, respectively, knowledge, monitoring, analysis, planning, and execution.

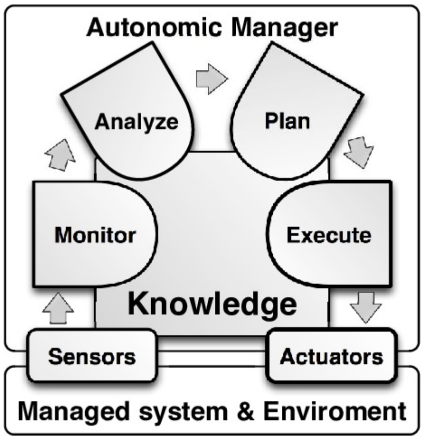

Figure 1. Mape-K Autonomic Control Loop [14].

- The Knowledge module is in charge of keeping relevant data in the memory to accelerate decision making.
- The Monitoring module uses sensors to collect data from the managed element, which could be a software or hardware resource, or an autonomic manager itself.
- The Analysis module provides mechanisms to interpret the collected data from the monitoring phase and predict future situations.
- The Planning module builds the necessary actions to achieve the goals.
- The Execution module uses effects to make changes on managed elements.

## IV. ROUTING PROTOCOL FOR LOW POWER AND LOSSY NETWORKS

RPL is an IPv6 routing protocol for Low power and Lossy Networks (LLNs) that specifies how to build a Destination Oriented Directed Acyclic Graph (DODAG) using an objective function and a set of metrics and constraints to compute the best path [15].

The RPL differs from other routing protocols that operate in less-constrained environments. In LLNs, especially when the network is made of devices that must save energy, it is imperative to limit the control plane traffic in the network.

The graph built by RPL is a logical routing topology built over a physical network to meet a specific criteria and the network administrator may decide to have multiple routing topologies (graphs) active at the same time used to carry traffic with different set of requirements [15].

## V. ATTACKS IN INTERNET OF THINGS

Most of threats in IoT environment attack the limited power of the sensors. The limited power of these devices exposes the network to many threats [16]. In this section, we will discuss some of the latest and more common attacks on the environment of the IoT and wireless sensor networks [17].

*1) Selective forward:* In a Selective Forward attack, the attacker node receives the transmission packets, but refuses to transmit some of them and drops those that it refused to transmit. The attacker must choose which packets to discard according to some standard such as size, destination or origin [12]. In this case, only the packets released by the attacker node can be freely transmitted.

*2) Black Hole:* In a Black Hole attack, the attacker node receives the transmission packets and drops all packets received, regardless of type, size, origin or destination [12].

*3) Sinkhole:* In a Sinkhole attack, the attacker tries to attract all the traffic from neighboring nodes [18]. So, practically, the attacker node listens to all data transmitted from neighboring nodes. Only this attack does not cause too much damage in the network, but together with another type of attack (Selective Forward or Black Hole), can become very powerful.

*4) Flooding:* In a Flooding attack, the attacker explores the vulnerabilities related to the depletion of the memory and the energy. One manner to take advantage of this vulnerability is when an opponent sends too many requests trying to connect to the victim, every request makes the victim allocate the resources in an attempt to maintain the connection [19]. Thus, to prevent the total resource depletion is necessary to limit the number of connections. However, this solution also prevents valid nodes to create a connection with the victim, causing problems such as queuing [19].

*5) Hello Flood:* In a HelloFlood attack, the attacker uses a device with a powerful signal to regularly send some messages; that way, the network is left in a state of confusion [17]. In order to find ad-hoc networks, many protocols use Hello Messages for discovering neighbor nodes and automatically create a network. With the Hello Flood attack, an attacker can use a device with high transmission power to convince every other node in the network that the attacker is its neighbor, but these nodes are far away from the attacker. In this case, the power consumption of sensors is significantly increased, because of protocols that depend on exchange information between neighbor nodes for topology maintenance or flow control [17].

Previously, we saw some of the most common attacks on IoT networks and, next, we analyzed the possible strategies to end or to mitigate the damage caused by them.

To stop the damages caused by attacks on a network, first it is necessary to detect these attacks, using an IDS. An IDS analyzes network activity and attempts to detect any unusual behavior that may affect the integrity of the network. Based on information provided by IDS, strategies are created to cope with the attacks. For example:

- To mitigate Sinkhole - If the geographical locations of the nodes of RPL DODAG are known, the effect of Sinkhole attacks can be mitigated by the use of flow control, making sure, that the messages are traveling to the correct destination. The RPL protocol also supports multiple instances DODAG offering alternative routes to the root DODAG [20].
- To mitigate Hello Flood - A simple solution to this attack, it is perform a bidirectional check for each message ”HELLO” [21]. If there is no recognition, the path is assumed to be bad and a different route is chosen. If geographical locations of the nodes of RPL DODAG are known, all packets received from a node that is far beyond of the common network node transmission capacity can be dropped.
- To mitigate Selective Forward - An effective countermeasure against Selective Forward attacks is to ensure that the attacker cannot distinguish the different type of packets, forcing the attacker to send all or none packets [22].

Raza et al. [11] indicated that the most efficient and fastest way to stop the damage of routing attacks is to isolate the malicious node. Some forms to ignore the attacker node were studied. These forms are:

- The Black List: After identifying the nodes and finding the attackers, a list will be created and all the malicious nodes will be added in order to exclude them from the possible routes of traffic data. To ignore the attacker, a verification will be done against the Black List excluding all nodes found of the typical RPL DODAG that have a root and multiple nodes.
- The Gray List: After identifying the nodes and finding the attacker, a list will be created. The suspicious attacker node will be added to this list with the intention of excluding it from the possible routes of traffic data, for a predetermined time. After the end of the predetermined time the suspicious attacker node is deleted from the list. In this way, if there is any doubt about the identity of the attacker node, the node may re-join the network. To ignore the suspicious attacker nodes, when creating the routing, a verification will be done against the Gray List excluding all nodes found of the typical RPL DODAG that have a root and multiple nodes.
- The White List: As in the example of the Black List, a list will be created after identifying the nodes and finding the attacker node. But, this time, will be add into the White List only the valid nodes and all malicious nodes will be excluded. This way will have a verification stating which nodes are valid and must belong to a typical RPL DODAG with a root and several nodes.

## VI. SELF-PROTECTION ARCHITECTURE

In Figure 2, we can see the self-protecting architecture for IoT proposed in this research. It consists of five modules (Monitoring, Analysis, Planning, Executing and Knowledge) and it is based on the MAPE-K autonomic control loop [13].

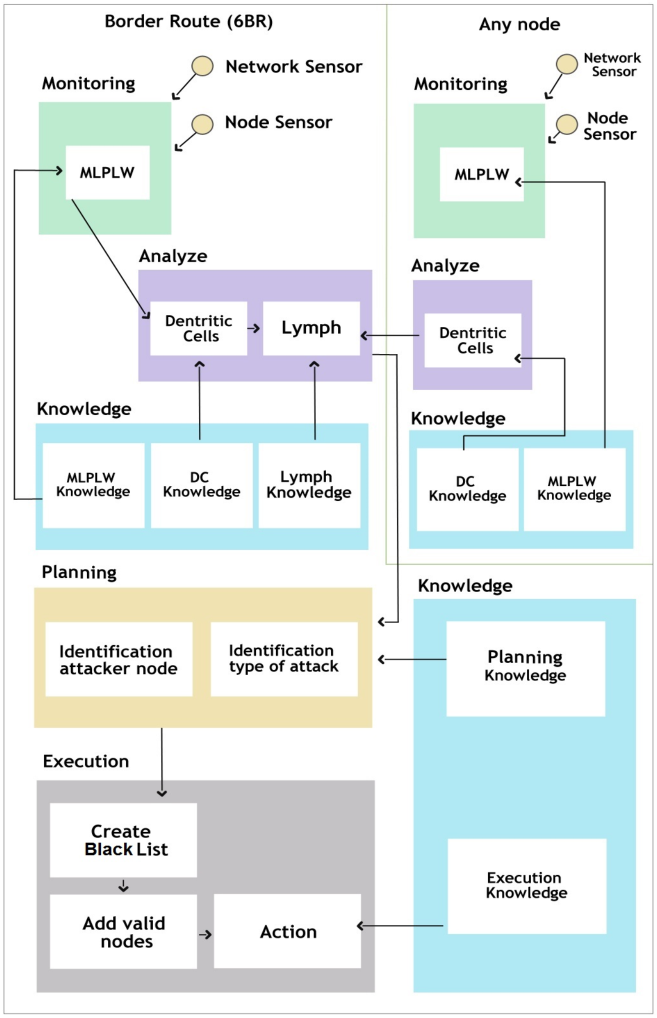

Figure 2. The Self-Protecting Architecture.

The monitoring and analysis modules are responsible, respectively, for collecting through the sensors, some information of the network that will be analyzed to measure the possibility of being associated with an attack. These two modules are present in the network nodes and in the border router (6BR).

The planning and execution modules will be responsible, respectively, for identifying the attacker, the type of attack and to mitigate the damage in the network. The information listed as relevant data for analysis, planning and execution is: type of transport protocol, type of application protocol, time of communication, number of messages sent, number of messages effectively sent and number of messages received.

The components of knowledge phase will be responsible for keeping all the knowledge acquired by the system. Knowledge about the planning and execution modules will be at 6BR. The information kept by the Knowledge Module will be used to facilitate and accelerate the discovery of the type of attack, the attacker node and the action to be taken to protect the network.

The complete design of this architecture can be found in Mello et al. [23]. The monitoring and analysis phases were implemented in [24]. Now, we describe how we implement the execution phase.

## VII. Execution Phase

The component responsible for the execution phase should mitigate or stop the damage caused by the attacks occurred on the network. The type of attack and the identification of the attacker node will be the information that will influence in the choice of the predetermined action to mitigate or stop the damage in the network. These two important pieces of information will be provided by the Planning Phase. The reason to find out the attacker node is, trying to isolate as quickly as possible and create a new route, thus, avoiding further damage to the network. The type of attack will be among one of the two groups mentioned in Figure 3. According to the group selected there will be a specific action to solve the problem, because each type of attack causes different types of damage on the network.

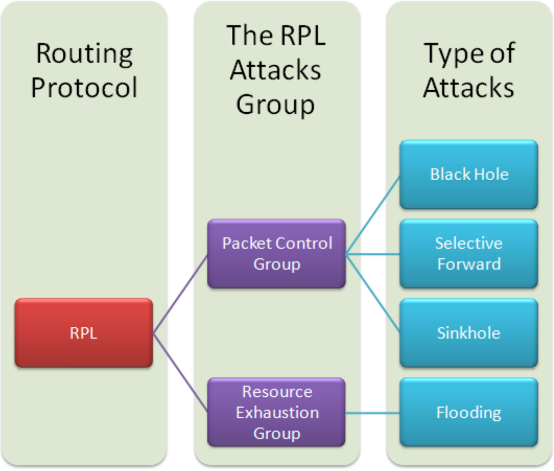

Figure 3. Taxonomy of RPL attacks

The first action to be taken by the components of the execution module, after the attack is detected, it is to ignore the malicious node. To perform this action it is important to identify the network nodes as legitimate or malicious. Raza et al. [11] say that it is necessary to be careful with the way of identifying nodes. If possible, it should be avoided the identification by IP address or MAC address, because they can be easily falsified. After making the identification of the nodes, some ways to ignore the attacker node were studied. Three ways to isolate the malicious node were analyzed before choosing the most convenient. The three ways are: Black List, Gray List and White List.

The way chosen was the Black List, because the maintenance of this list is simple. This way, all valid nodes will recalculate their rank in the RPL protocol (DODAG). To recalculate the rank of all valid nodes, it will be necessary to ignore the DODAG Information Object (DIO) of all nodes with higher rank than theirs and the DODAG Information Solicitation (DIS) and DIO of nodes that are present in the Black List. Thus, for a stranger node to join the network, it should be reported as safe and not be present in the Black List.

## VIII. Results

It was possible to simulate the Flooding and Black Hole attacks (with SinkHole and Selective Forward variants). The simulations with and without the attacking node were performed in the Cooja, a simulator of the ContikiOS, following a DODAG topology in which there is a certain number of nodes and one of them will be the root.

We defined the number of nodes in the simulation and all used the same platform (Skymote). The routing protocol used was the RPL and the addressing protocol used was the IPV6.

It should also be noted that the simulation time should be long enough for the data collection to begin. In our simulation, we used a virtual time of 2 minutes. In the simulations, we used 11 nodes with a transmission rate of 50 meters and interference range of 100 meters. The network simulation was generated from the Cooja Simulator and can be seen in Figure 4.

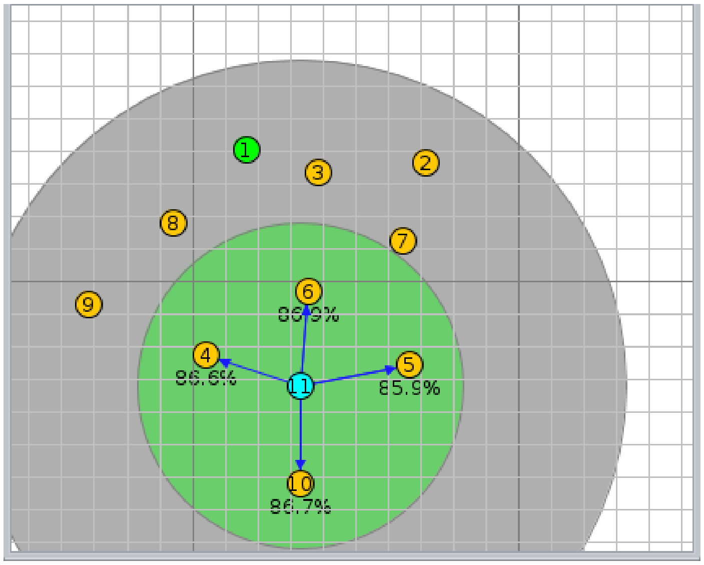

Figure 4. Network Simulation.

After running the simulation for 2 minutes (virtual), the Directed Acyclic Graph (DAGs) was generated by the Cooja Simulator looks as shown in Figure 5. The node with ID 1 is the root and the node with ID 11 seen in Figure 4 and Figure 5, at first, is a common valid node, thus enabling the simulation of the network without the presence of attacks. But to simulate

ISBN: 978-1-61208-562-3

the presence of the attacks on the network the node with ID 11 has been modified to act as the malicious node.

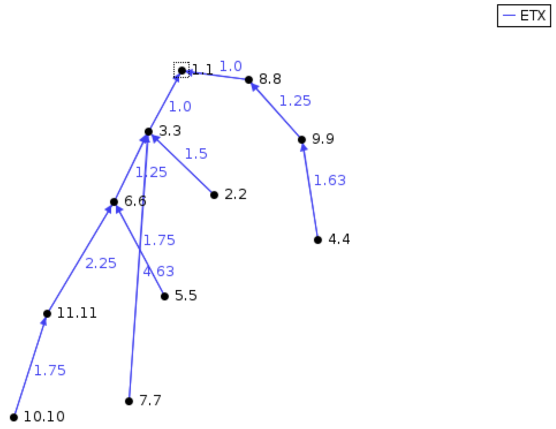

Figure 5. The Directed Acyclic Graph of the simulated network.

### A. The simulation without attack

In this simulation, we do not have attacks and all nodes are valid. As can be easily seen in Figure 6, all nodes have the consumed power of less than 10%.

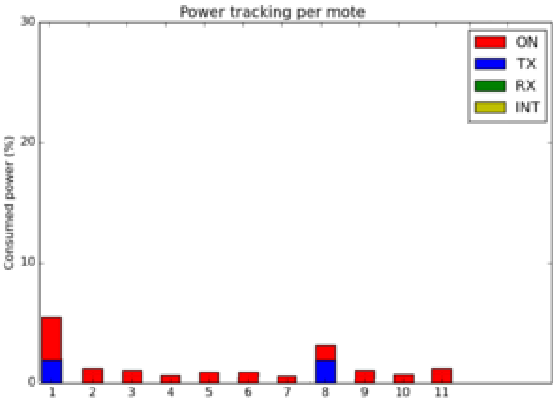

Figure 6. The Consumed Power without attack.

### B. The simulation of the Black Hole attack

In this simulation the malicious node dropped all the collected data application messages instead of forwarding them. When using the Selective Forward variant in the simulation, only the received data plane messages of some nodes with IP specified by the attacker node were dropped. This way, it was easily observed a malfunction in the network in relation to packets delivery and packet integrity.

The Black Hole attack can also be enhanced if combined with a sinkhole attack. When simulating the Black Hole with the sinkhole variant, the DAG was changed. Some valid nodes (ID 4, 5 and 10) in the neighborhood of the malicious node (ID 11) have now set it as their parent. This way, the attack has become even more efficient, because it is listening and dropping a larger number of the received messages. The DAG changed can be seen in Figure 7 generated from the Cooja Simulator.

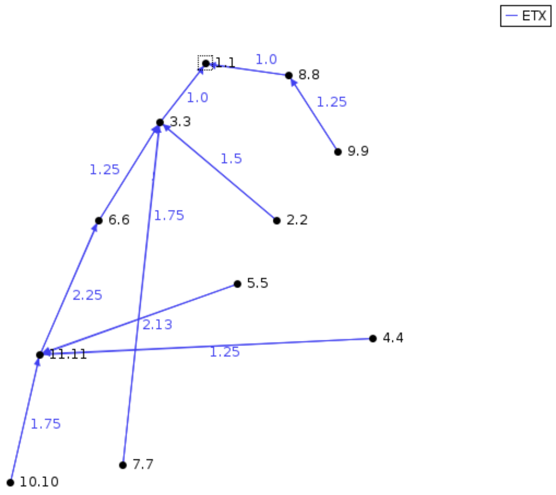

Figure 7. The DAG of the simulated network with Sinkhole attack.

### C. The simulation of the Flooding attack

In this simulation the malicious node (Node with ID 11) impacts nodes with IDs 4, 5, 6 and 10 (Figure 4). It is very easy to see in Figure 8 that these nodes are particularly affected by the attack in terms of ON and RX times and the malicious node consumed a lot power with TX. So these nodes spend a lot of energy and memory, to read the requests sent by the malicious node. The power consumed by the attacker node and by the nodes affected by the attack is well over 10% but, the power consumed by other nodes remains below 10%.

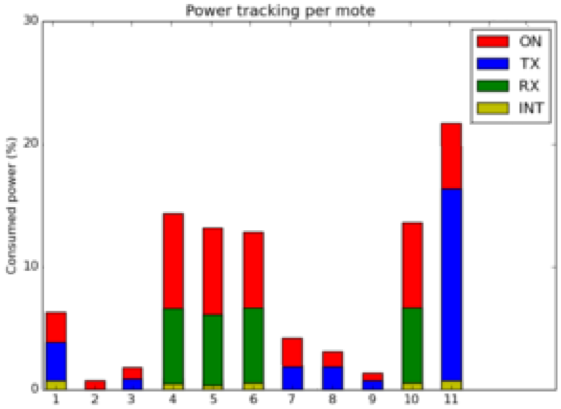

Figure 8. The Consumed Power with Flooding attack.

### D. The simulation with our Architecture to isolate the attacker node

During the execution phase, our architecture is intended to isolate the attacker node so that it does not cause further damage to the network. When simulating the isolation of the attacker node, the DAG was changed and another node, besides the malicious node (ID 11) was also isolated.

The other node also isolated can be seen in Figure 9. This other node was the with ID 10. This occurred because the node with ID 10 was very far from the others.

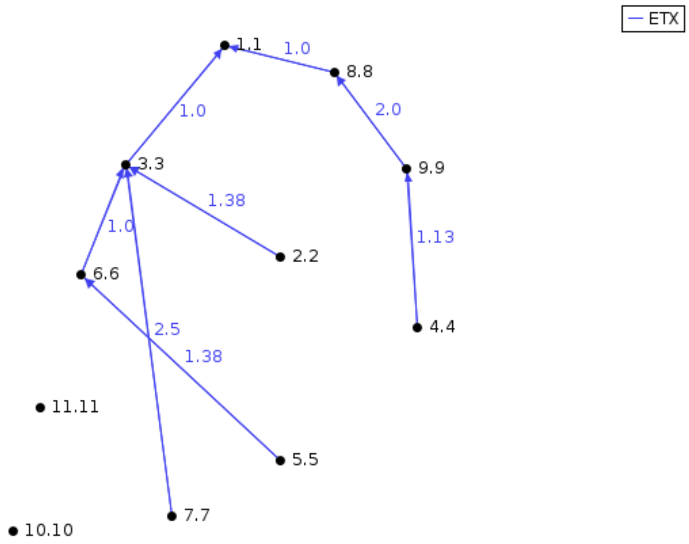

Figure 9. The DAG of the simulated network.

## IX. CONCLUSION

In this paper, we have used the MAPE-K control loop to design a new self-protecting architecture for the IoT. With this architecture, it will be possible to incorporate security facilities in the IoT systems releasing the programmer to treat only the functional requirements. Besides, the user can extend the architecture developing new security services to handle specific attacks.

The research will help provide a self-protection mechanism for IoT networks, facilitate the detection and the classification of possible attacks on smart devices, mitigate the damage of the attacks suffered ensuring better performance and increasing the confidence of users when using devices connected to IoT network.

The Self-Protecting architecture deals with five different attacks (Sinkhole, Selective forward, Black Hole, Flooding and Hello Flood) bearing in mind the memory consumption and energy due to a lack of resource of the available devices in the IoT environment.

The performance of the system should be evaluated to verify if the Self-Protecting architecture has better results than related work. New attacks and new technologies will emerge, and then the work presented here may be extended to address those.

## ACKNOWLEDGMENT

This work was supported by CAPES and FAPITEC/SE

## REFERENCES

[1] D. INFSO, “Internet of things in 2020: Roadmap for the future,” INFSO D, vol. 4, 2008.

[2] D. Niyato, E. Hossain, and S. Camorlinga, “Remote patient monitoring service using heterogeneous wireless access networks: architecture and optimization,” IEEE Journal on Selected Areas in Communications, vol. 27, no. 4, 2009.

[3] O. Vermesan, P. Friess, and A. Furness, “The internet of things 2012: New horizons,” IERC 3rd edition of cluster book, 2012.

[4] J.-P. Vasseur and A. Dunkels, Interconnecting smart objects with ip: The next internet. Morgan Kaufmann, 2010.

[5] M. R. Nami and M. Sharifi, “Autonomic computing: a new approach,” in Modelling & Simulation, 2007. AMS’07. First Asia International Conference on. IEEE, 2007, pp. 352–357.

[6] J. Kephart, G. Sorkin, M. Swimmer, and S. White, “Blueprint for a computer immune system,” in Artificial immune systems and their applications. Springer, 1999, pp. 242–261.

[7] S. R. White, M. Swimmer, E. J. Pring, W. C. Arnold, D. M. Chess, and J. F. Morar, “Anatomy of a commercial-grade immune system,” IBM Research White Paper, 1999.

[8] G. Portokalidis and H. Bos, “Sweetbait: Zero-hour worm detection and containment using low-and high-interaction honeypots,” Computer Networks, vol. 51, no. 5, 2007, pp. 1256–1274.

[9] M. Swimmer, “Using the danger model of immune systems for distributed defense in modern data networks,” Computer Networks, vol. 51, no. 5, 2007, pp. 1315–1333.

[10] S. Rawat and A. Saxena, “Danger theory based syn flood attack detection in autonomic network,” in Proceedings of the 2nd international conference on Security of information and networks. ACM, 2009, pp. 213–218.

[11] S. Raza, L. Wallgren, and T. Voigt, “Svelte: Real-time intrusion detection in the internet of things,” Ad hoc networks, vol. 11, November 2013, pp. 2661–2674.

[12] D. M. Shila, Y. Cheng, and T. Anjali, “Mitigating selective forwarding attacks with a channel-aware approach in wmns,” Wireless Communications, IEEE Transactions on, vol. 9, May 2010, pp. 1661–1675, doi:10.1109/TWC.2010.05.090700.

[13] J. O. Kephart and D. M. Chess, “The vision of autonomic computing,” Computer, vol. 36, January 2003, pp. 41–50, doi:10.1109/MC.2003.1160055.

[14] D. Weyns, S. Malek, and J. Andersson, “Forms: a formal reference model for self-adaptation,” in Proceedings of the 7th international conference on Autonomic computing. ACM, June 2010, pp. 205–214, doi:10.1145/1809049.1809078.

[15] T. Winter, “Rpl: Ipv6 routing protocol for low-power and lossy networks,” 2012.

[16] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” Computer networks, vol. 54, October 2010, pp. 2787–2805, doi:10.1016/j.comnet.2010.05.010.

[17] D. Martins and H. Guyennet, “Wireless sensor network attacks and security mechanisms: A short survey,” in Network-Based Information Systems (NBiS), 2010 13th International Conference on. IEEE, November 2010, pp. 313–320, doi:10.1109/NBiS.2010.11.

[18] P. Goyal, S. Batra, and A. Singh, “A literature review of security attack in mobile ad-hoc networks,” International Journal of Computer Applications, vol. 9, November 2010, pp. 11–15.

[19] A. D. Wood and J. A. Stankovic, “Denial of service in sensor networks,” Computer, vol. 35, December 2002, pp. 54–62, doi:10.1109/MC.2002.1039518.

[20] T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. S. Kumar, and K. Wehrle, “Security challenges in the ip-based internet of things,” Wireless Personal Communications, vol. 61, September 2011, pp. 527–542, doi:10.1007/s11277-011-0385-5.

[21] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: Attacks and countermeasures,” Ad hoc networks, vol. 1, September 2003, pp. 293–315.

[22] L. Wallgren, S. Raza, and T. Voigt, “Routing attacks and countermeasures in the rpl-based internet of things,” International Journal of Distributed Sensor Networks, vol. 2013, June 2013.

[23] R. de AC Mello, A. de RL Ribeiro, F. M. de Almeida, and E. D. Moreno, “An architecture for self-protection in internet of things,” ICWMC 2016, 2016, p. 51.

[24] F. M. de Almeida, A. d. R. L. Ribeiro, and E. D. Moreno, “An architecture for self-healing in internet of things,” UBICOMM 2015, 2015, p. 89.