AUTONOMIC TRUST MANAGEMENT IN CLOUD-BASED AND HIGHLY DYNAMIC IOT APPLICATIONS

Suneth Namal , Hasindu Gamaarachchi* , Gyu Myoung Lee** , Tai-Won Um****

Department of Computer Engineering*
University of Peradeniya , P.O. Box 20400 , Sri Lanka
Department of Computer Science**
Liverpool John Moores University , Liverpool , United Kingdom
Broadcasting & Telecommunications Media Research Laboratory***
Electronics and Telecommunications Research Institute, Daejeon, Korea
Email: namal@ce.pdn.ac.lk* , hasindu2008@gmail.com* , g.m.lee@ljmu.ac.uk** , twum@etri.re.kr***

ABSTRACT

In this paper, we propose an autonomic trust management framework for cloud based and highly dynamic Internet of Things (IoT) applications and services. IoT is creating a world where physical objects are seamlessly integrated in order to provide advanced and intelligent services for human-beings in their day-to-day life style. Therefore, trust on IoT devices plays an important role in IoT based services and applications. Cloud computing has been changing the way how provides are looking into these issues. Many studies have proposed different techniques to address trust management although non of them addresses autonomic trust management in cloud based highly dynamic IoT systems. To our understanding, IoT cloud ecosystems help to solve many of these issues while enhancing robustness and scalability. On this basis, we came up with an autonomic trust management framework based on MAPE-K feedback control loop to evaluate the level of trust. Finally, we presents the results that verify the effectiveness of this framework.

Index Terms— Trust, Internet of Things, Cloud Networks, IoT cloud ecosystem, smart homes, MAPE-K

1. INTRODUCTION

One of the preliminary objectives of Internet of Things (IoT) is to deliver personalised or even autonomic services to individuals, building on a pervasive digital ecosystem that collects information and offers control over devices that are embedded in every one of ours' everyday lives [1]. The extraordinary power of this vision is expected to lead to fundamental social change: it will affect the way in which we interact with our environment and each other, and will result in the creation of new business opportunities and innovative business models [2]. However, the embedded nature of this technology and a lack of awareness of its potential level of trust and, social and personal consequences, as balanced against the more clearly articulated benefits, makes specific issues correspond to trust, security and privacy [3]. On the other hand, cloud computing has virtually unlimited capabilities in terms of storage and processing power, is a much more mature technology, and has most of the IoT issues at least partially solved.

The IoT integrates a large amount of everyday life devices from heterogeneous network environments, bringing a great challenge into trust, security, and reliability management. In doing that, smart objects with heterogeneous characteristics should cooperatively work together [4]. It is a known fact that the Devices in IoT very often expose to public areas and communicate through wireless, hence vulnerable to malicious attacks [5, 6, 7]. Migrating IoT application specific data into the Cloud offers great convenience, such as reduction of cost and complexity related to direct hardware management [8, 9, 10]. However, to evaluate the trustworthiness of their systems cannot use only the past experiences, since the novel autonomic systems nowadays are highly dynamic and the behaviors are unpredictable. These restrictions are detrimental to the adaptation of Trust Management Systems (TMSs) to today's emerging IoT architectures, which are characterized with autonomic and heterogeneous nodes and services.

Clouds or cloud computing has picked up many researchers' attention, as such it is being a part of IoT. Undoubtedly, trust management is the most challenging issues in emerging cloud systems where millions of services, applications and nodes deployed together under a single umbrella to serve each other [11]. Together with the current dynamism of the systems and the autonomous users' behavior, the latter task has been too complicated [12]. In reality, autonomic trust management is hard to be realized because the cloud of things is hard to control due to the scale of deployment, their mobility and often their relatively low computation capacity [13, 14]. As a result, the trust manager itself should be adaptive to the autonomic conditions posed by the system.

In this paper, we propose a framework for autonomic trust

This research was supported by the ICT R&D program of MSIP/IITP [R0190-15-2027, Development of TII (Trusted Information Infrastructure) S/W Framework for Realizing Trustworthy IoT Eco-system].

management based on Monitor, Analyse, Plan, Execute, Knowledge (MAPE-K) feedback loop to evaluate the level of trust in a IoT cloud ecosystem. Even though many research activities were carried-out in the scope of autonomic trust management, non of them have addressed how an integration between IoT and cloud would work. We utilize MAPE-K feedback control loops to enhance consistency of the system while improving robustness and scalability with the introduction of cloud concepts.

The rest of the paper is organized as follows; Section 2 describes the related work. Section 3 describes challenges of TMSs in IoT, next Section 4 describes cloud integration in IoT, Section 5 presents the system model and Section 7 describes simulation and results. Finally, in Section 8, we conclude the paper.

2. RELATED WORK

Yan et al. [15] have done a survey on trust management for IoT where they discuss the current state of art while elaborating open issues and key challenges in IoT trust management. They have categorized trust properties into five categories and proposed ten objectives for trust management in IoT. Manuel [16] introduces a trust model for a cloud resource provider where they use four parameters namely availability, reliability, turnaround efficiency and data integrity for the evaluation of trust. Their model is based on the present value as well as the history of the parameters.

Chen et al. [17] have created a trust model for IoT that uses fuzzy sets. They focus mainly on different security challenges such as detecting malicious attacks. Firdhous et al. [18] have done a critical review of trust management in Cloud Computing. They discuss existing TMSs for the cloud and compare them based on a set of parameters. Noor et al. [19] have introduced a framework for trust management in cloud environments called Trust as a Service. Their work helps to differentiate credible trust feedbacks from malicious feedbacks where feedbacks originate from consumers of the cloud service.

3. CHALLENGES OF TRUST MANAGEMENT IN IOT

The current IoT systems challenge TMSs in following different aspects. First, the behavioral features of an IoT system is expected to have huge amount of entities. The problem with that is the existing trust management protocols do not scale well to accommodate this requirement because of the limited storage and computation power. Second, an IoT system evolves with new applications, services, and nodes frequently joining and leaving the systems. Therefore, a trust management protocol must address this issue at the same time in order to allow newly joining elements to build up trust quickly with a acceptable level of accuracy.

Third, the building blocks or entities of IoT systems are mostly human carried or human operated devices, which implies that a TMS must be capable of compensating the human errors at some level. At this point that IoT may take into account the social relationships among entity owners in order to maximize protocol performance. Lastly and arguably most importantly, like other Internet systems, an IoT system is frequently the target of many cyber attackers, since many IoT entities are accessible through wireless networks, the network itself is a point of failure in terms of the level of trust offered. Therefore evaluating the level of trust in such autonomic and hostile environments has been a critical challenge.

4. CLOUD INTEGRATION IN INTERNET OF THINGS

Even though the worlds of cloud computing and IoT seem to evolve independently on their own paths, an integration of Clouds with IoT will lead to the production of large amounts of data, which needs to be securely stored, processed and accessed. Cloud computing as a paradigm for big data storage and analytic needs the trustworthiness. Cloud can benefit from IoT by extending its scope to deal with real world things in a more distributed and dynamic manner, and for delivering new services in a large number of real life scenarios. Essentially, the Cloud acts as intermediate layer between the things and the applications, where it hides all the complexity and the functionalities necessary to implement the latter.

Trust is one of the most concerned obstacles for the adoption and growth of cloud computing. Although couple of solutions have been proposed, determination of credibility of trust feedbacks is neglected in most of the cases which lead to many security failures. TMSs usually experience malicious behaviors from its users. In addition, managing trust feedbacks in cloud environments is a difficult problem due to unpredictable number of cloud service consumers and highly dynamic nature of cloud environments. On the integration of clouds with IoT, there are many advantages. Adoption of clouds enables new scenarios for smart services and applications. New scenarios for smart services and applications based on the extension of Cloud through the things delivers extensions in IoTs.

IoT is characterized by a very high heterogeneity of devices, technologies, and protocols. Therefore, scalability, interoperability, reliability, efficiency, availability, and security can be very difficult to obtain. Sensing as a service, sensing and actuation as a service, sensor events as a service, sensor as a service, database as a service, data as a service, and Ethernet as a service are all different potential extensions to IoT based clouds. IoT makes IP-enabled devices communicate through dedicated hardware, where the support for such communication can be expensive.

The applications of such systems could be extended to healthcare, smart cities, smart homes and smart metering, smart grids, etc. However, so far non of the research activities has been carried out in the scope of trust management in cloud integrated IoT. Automation of management is one of the essential characteristics of the cloud networks today.

Autonomic computing is an approach to equip computer systems with capabilities to autonomously adapt their behavior and/or structure according to dynamic operating conditions. For effective self management, a system needs context awareness, self-configuration, self-optimization, self-protecting, self-management, self-healing, anticipatory, and openness.

5. SYSTEM MODEL

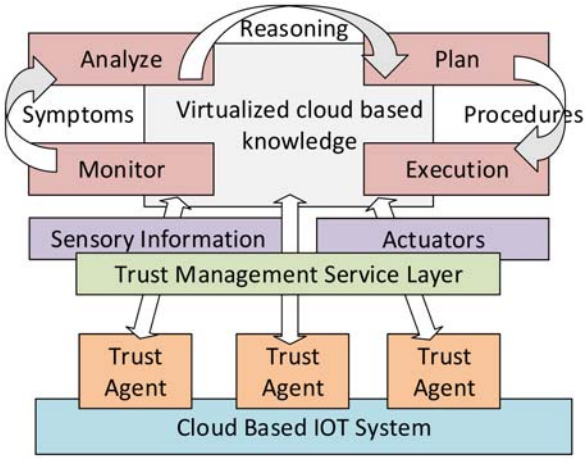

Figure **1**. MAPE-K feedback loops for adaptive trust agents.

The system we are interested is highly dynamic which implies the need for adaptive decision making and autonomic agents with control loops to manage resources. A promising approach to handle such dynamics is self-adaptation that can be realized by a MAPE-K feedback loop. To provide an evidence that the system goals are satisfied, regarding the changing conditions, state of the art advocates the use of formal methods. However, it is important to remark that the trust agents in Fig. 1 do not replace the monitoring phase of the MAPE-K, but instead it filters out the trust information from other information while holding the required knowledge to support the autonomic decision-making process.

The distributed nature of the trust agents assure quick responses and scalability of the solution. In Fig. 1, the monitor function aggregates, correlates and further filters the information until it determines a symptom that needs to be analyzed. Analyze function performs complex data analysis and reasoning on the symptoms provided by the monitor function. Analyze function would be influenced by stored knowledge data which, in fact, virtually centralized but physically exists within the trust agents. If changes are required, a change request is logically passed to the plan function. The plan function structures the actions needed to achieve goals and objectives and creates or selects a procedure to enact a desired alteration in the managed resource. At the same time it can take on many forms, ranging from a single command to a complex work-flow. Execution phase changes the behavior of the managed resource using effectors, based on the actions recommended by the plan function. In fact, the executors are open APIs to the trust managers' feedback system. The knowledge in Fig. 1 is the standard data associated with the monitor, analyze, plan and execute functions. The knowledge here is shared among the trust agents and could be virtually centralized using cloud techniques to facilitate decision making. This would include data such as all trust related information, context information, topology information, historical logs, metrics, symptoms, policies, etc. This system now becomes self-adaptive based on MAPE-K feedback loops that deal with dynamic trust issues arising due to openness. It is important to notice that our particular focus is on adaptations that require elevating or downgrading the level of trust in a system.

5.1. Trust as a Service (TaaS)

Cloud is a flexible framework to effectively implement services. Among many other services "Trust" can be thought of as a service offered by the cloud system to its users. In an IoT system, multiple devices would associate with each other as well with users. Internet and IoT play a significant role in service deployment, especially in facilitating and automating the human needs and requirements. An effective trust management system helps cloud service providers and consumers reap the benefits brought about by cloud computing technologies.

Despite the benefits of trust management, several issues related to general trust assessment mechanisms, distrusted feedbacks, poor identification of feedbacks, privacy of participants and the lack of feedbacks integration still need to be addressed. Traditional trust management approaches such as the use of Service Level Agreements (SLA) are inadequate for complex IoT based cloud environments. Sometimes, the vague clauses and unclear technical specifications of SLAs can lead cloud service consumers to be unable to identify trustworthy cloud services. For example, a smart home environment could be one of the possible applications of cloud based IoT system that implements services. Fig. 2 presents a smart home environment in which the proposed TMS could be applied. Nowadays, modern homes are equipped with many IoT devices that are automated and controlled remotely through Internet. For example, an owner may access the electrical devices at his home through his mobile device.

It could be like switching the security system on or monitoring through the surveillance cameras when he is away from his home. However, the risk behind such a solution is about producing the wrongful information that mislead the owner. For example, at the time the owner remotely switch on the home security system, a criminal may produce a faulty acknowledgment and send it to the owner to misguide him and burglarize. Because of that, the trust on IoT devices and their applications in real-world is critical. There have been many different approaches to enhance the trust over information and devices. These solutions address several trust related issues in common. They are;

- Trust plays a critical role in risky and uncertain environments that are not under control.

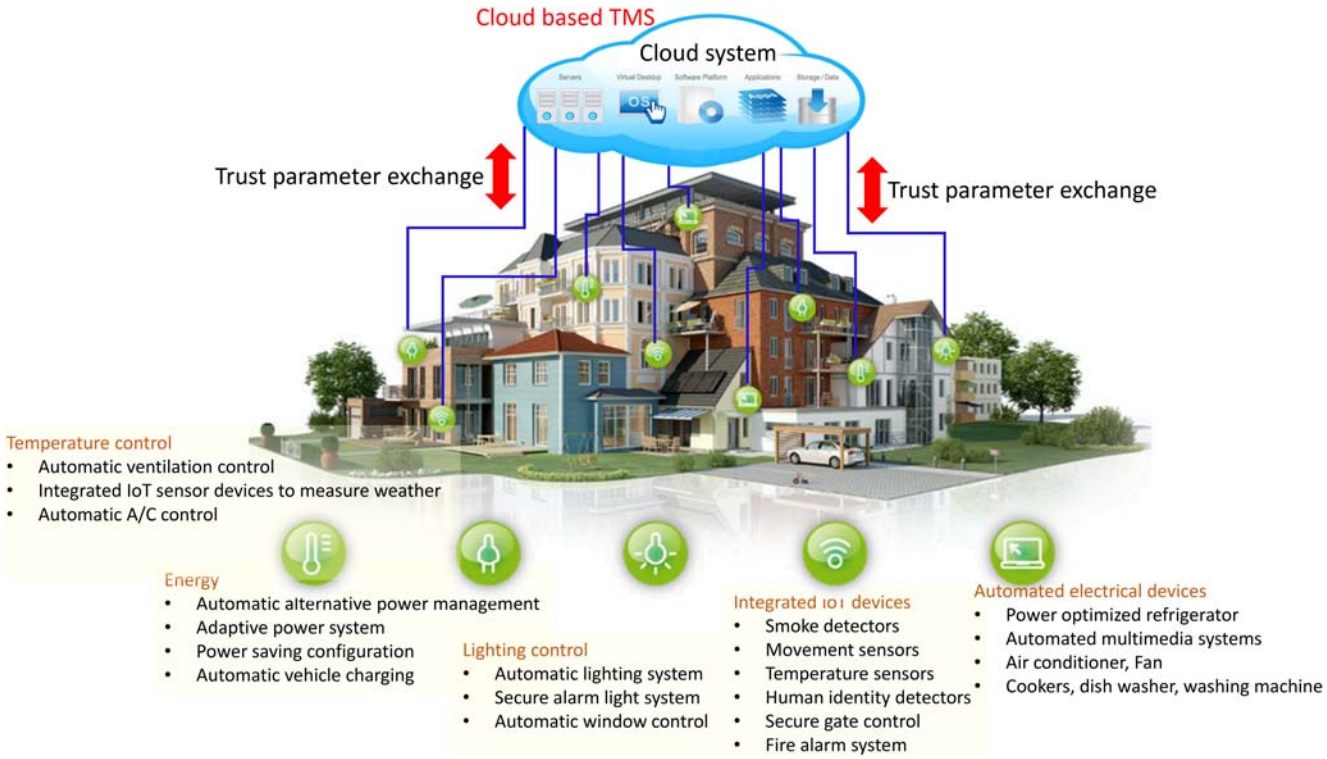

Figure **2**. Smart home environment with the trust management system. The IoT devices sense the trust parameters and exchange information to the trust agents virtualized in the cloud network.

- Trust is the basis on which certain decisions are made in our day-to-day life style.
- The decisions are taken mostly on the prior experience and knowledge where wrongful history produces incorrect results.
- Trust is subjective and it is based on the personal opinion and their preferences.
- Dynamic environmental and contextual information modifies level of trust. Possible changes with time and new knowledge may override influence over the old ones.
- Trust is context-dependent which may produce incorrect information in extreme contextual conditions.

5.2. Cloudifying TaaS

The TMS proposed in this paper acquires the contextual and environmental information through the IoT sensor devices and deliver it to the trust agents which filter information and send it to the MAPE-K control loop implemented on the cloud. Therefore, trust now operates as a service on top of the cloud, which we call "TaaS". Behind cloudfying the application, many advantages would be delivered to the end users.

- Availability: availability of the "TaaS" service could be thought as the reachability between the target environment and the cloud system. "TaaS" will communicate with the IoT devices and sense the information.

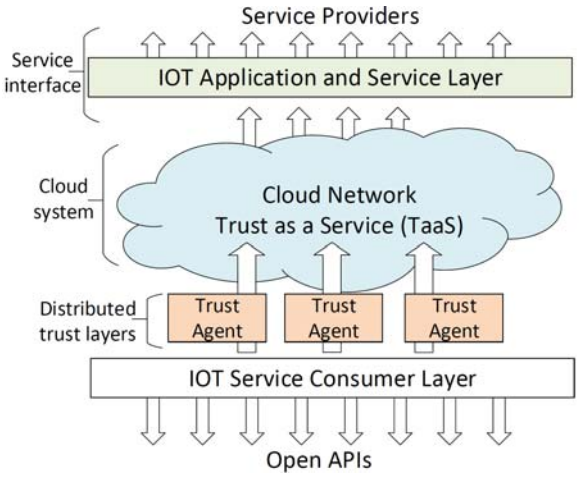

Figure **3**. Overview of the solution architecture.

As far as the devices are connected to Internet, service will always be available to its users.

- Scalability: scalability defines the ability of "TaaS" to handle the growing number of IoT devices. Over the time, many houses hold smart devices would be added to the Internet. In order to cater them, "TaaS" defines distributed trust agents that filter raw data.
- Accessibility: in relation with our example of smart home environment, the user may need to switch on his home security system. As far as "TaaS" is imple-

mented on a cloud network, the user may access the service through Internet from wherever he is.

- Flexibility: MAPE-K control loop in the cloud aggregates trust parameters through the trust agents. The cloud system enables the trust agents to be deployed in a flexible and distributed manner. By doing so, the system allows the IoT devices to communicate trust related information.

Fig. 3 describes the solution architecture of the proposed trust management system that consists of distributed trust agents. They produce the trust parameters and filters them to the adaptive trust parameter pool which is on the cloud. The service consumer layer integrates the clients to the TMS. This layer consists of several distributed TMS nodes that expose interfaces to the clients. Cloud deploys trust as a service together with the MAPE-K control loop. The feedback system produces results based on the past history. In doing that, we normalize the impact of favorable abnormalities to reduce the expected dynamism. This is because the context has a significant affect on the level of trust. The raw information as it is will happen to produce incorrect decisions which we overcome with MAPE-K control loops.

6. SYSTEM MODEL

The systems model consists of three layers, service consumer layer, cloud network layer, and applications and service layer. Furthermore, the service consumer layer consists of open Application Programmable Interfaces (APIs) on which clients access the services and trust agents that locally filters trust related information to the trust data pool. In second layer - cloud network is implemented with the service ("TaaS") which utilizes cloud based computing intelligence to obtain the corresponding parameters. These parameters are then fed to the MAPE-K feedback control loop that produces the set of trust parameters on which the final decision is made. However, the process runs over many iterations to modify a final result based on the past history. Fig. 4 demonstrates this control loop which modifies the current level of trust and make decisions. In fact, we consider four trust related parameters; availability, reliability, response time and capacity.

- Availability is about making the resources available for users. The trustworthiness of a system lies on whether the resources are available when it is required.
- Reliability defines the level of trust among two entities. A reliable system always produces correct information.
- Irregularities in response time predicts possible intrusions in the system. That helps to identify changes from normal.
- Finally, capacity contributes to the model by assuring accessibility in one hand and scalability on the other hand.

"TaaS" measures the level of trust in terms of these parameters and finally aggregate them to make the decisions or to continue into a feedback loop that modifies current value based on past values. It is done by the analyzer which runs multiple computing methods for reasoning based on the provided set of parameters. "Planner" transforms reasoning to procedures which could be directly forwarded to the executors or adapters through which they are converted to decisions. If the decision does not fit in the context to be executed that can be modified with the past history and return it to another feedback loop for modifying. At last, all the parameters are stored in the adaptive parameter pool on the cloud and accessible by the service providers through application and service layer.

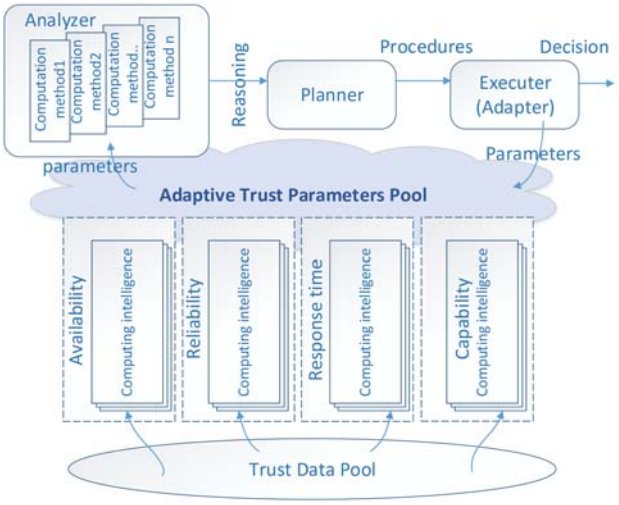

Figure **4**. State of art of trust agent.

This framework thus provides the flexibility on both clients and operators and adjust the trust level according to the context. The IoT sensor devices send raw data that they collect where their representation would differ with respect to which trust parameter they are contributing to. For example, a sensor that senses the availability would sense the number of successful ping requests made in a unit time interval while a sensor that senses reliability would measure the Bit Error Rate (BER) in the target environment.

As a result, the raw data must be first normalized and transformed appropriately to generate the trust value that depicts the current state with respective to the trust parameter. Since this value only describes the present status, we made an extension to that to integrate it with the previous history appropriately. In a nutshell, this complete process, where we represent the state as a combination of both history and the current state can be thought of as a feedback loop. Thereby, the MAPE-K control feedback loop comes into the model. Herein, a trust parameter for a IoT device d at a time t can be evaluated using Eq. (1).

$$P_{d,t} = (\alpha P_{d,t-1} + (1-\alpha) C_{d,t})^{\frac{1}{b}} \quad (1)$$

$$C_{d,t} = \left[\frac{-s(V_0 - V_{d,t})}{V_0 - V_{min}} r_1 + \frac{s(V_{d,t} - V_0)}{V_{max} - V_0} r_2 \right] \quad (2)$$

Thus, the same parameter at time $t-1$ is represented as $P_{d,t-1}$. $C_{d,t}$ represents the current value for the trust obtained via transformation of the sensor raw data using Eq. (2). In Eq. (1), α is the weight given on the history which should be a value between 0 and 1. The value b is a parameter that defines by how much the calculated trust values are to be augmented or diminished. A value slightly greater than 1 would result in an augmented trust level while a value slightly less than 1 would result in a diminished trust value. This parameter could be set based on the dynamic nature of the system such that the effect on the trust due to the high variations are compensated. In Eq. (2) parameters of the format V_k are with respect to the raw data that is received from IoT sensor. The maximum raw data value that can be generated by the sensor is given by V_{max} while the minimum is given by V_{min}. The raw data value update sent by a sensor about the device d at a time t is given by $V_{d,t}$.

The $P_{d,t}$ value evaluated by Eq. (1) always falls between -1 and 1. It reaches 1 at the highest level of trust while the value will remain around 0 when there is no trust on the device (the fact that device is neutral with respect to others). At the same time, it remains at 0 when there is no data to evaluate the trust, for example when a device is just added to the system. In case, if the device is untrustworthy and may cause a damage on others, then the value would reach -1. V_0 determines the value coming from the sensor that would result a zero value when normalized and transformed. A value $V_{d,t}$ which is less than V_0 can lead to a negative value for the trust parameter. The value for r_1 and r_2 in Eq. (2) must be selected as follows:

- if $V_{d,t} < V_0$ then $r_1 = 1$ and $r_2 = 0$
- if $V_{d,t} >= V_0$ then $r_1 = 0$ and $r_2 = 1$

The value for s must be selected based on the fact whether the raw data values received from the sensor is directly proportional or inversely proportional to the trust. When the parameter is directly proportional with trust, for example the number of successful ping request, s should be positive, i.e. 1. When it is inversely proportional to trust, for example the BER, then s should be negative, i.e. -1.

The formula we discussed so far only calculates a single trust parameter. The total trust of a system actually depends on multiple such parameters. Therefore, finally all the different parameters evaluated using Eq. (1) must be integrated to evaluate the effective trust level. Total trust can be evaluated by using weighted sum as given in Eq. (3). Here the effective total trust for a device d at time t denoted by $T_{d,t}$. A trust parameter calculated using Eq. (1) is depicted as $(P_{d,t})_i$ and there are n number of such different parameters. The respective weights assigned to each of those trust parameters are denoted by β_i.

$$T_{d,t} = \sum_{i=1}^{n} \beta_i (P_{d,t})_i \quad (3)$$

7. SIMULATION AND RESULTS

We simulate the proposed model for a smart home environment by using Matlab. We evaluate four parameters, namely availability, reliability, response time and capacity. First, for evaluating availability, we checked whether the devices are alive and reachable by sending out a fix number of ping requests. The number of responses thus, depends up on the route to the target device. Furthermore, any hardware failure will also happen not to receive any response. Then, evaluation of reliability was measured by simulating the possible BER on the target environment. Next, the response time was evaluated based on the round trip time whereas finally, the capacity was evaluated based on the number of current sessions on a device and the maximum number of connections to an IoT device.

The calculations are based on Eq. (1) and Eq. (2) discussed in previous section. The value for α for the trust parameters availability, reliability, response time and capacity was set to 0.8, 0.8, 0.8 and 0.9 respectively. The value for b was set to 1.16. Fig. 5 describes level of trust against availability, reliability, response time and capacity. The graphs with feedback demonstrate the level of trust when the trust protocol is applied. Without feedback demonstrates when it is not applied, where huge variation of level of trust can be seen over time. The significance here is the adaptation of MAPE-K control loop to improve the consistence of level of trust. That is because dynamic systems are highly vulnerable and may change their behavior/level of trust quite fast. To comply these needs, the framework applies history across the MAPE-K control loop in order to reduce impulses that misguide the TMS.

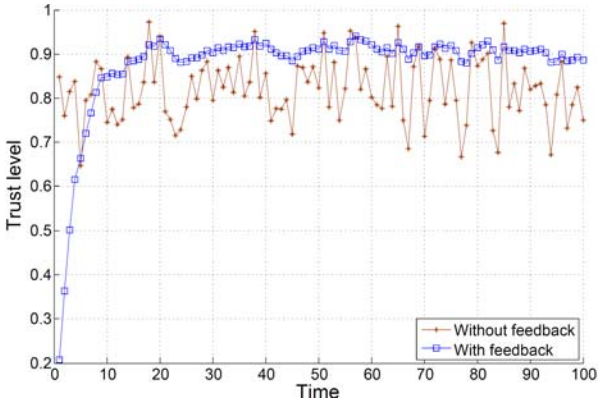

Figure **6**. Effective level of trust. (Aggregated availability, reliability, response time and capacity)

In doing so, we make sure the framework will not produce incorrect decisions at last. Finally, we integrate these trust parameters to obtain the effective level of trust. Fig. 6 presents the effective level of trust on which the decisions would be

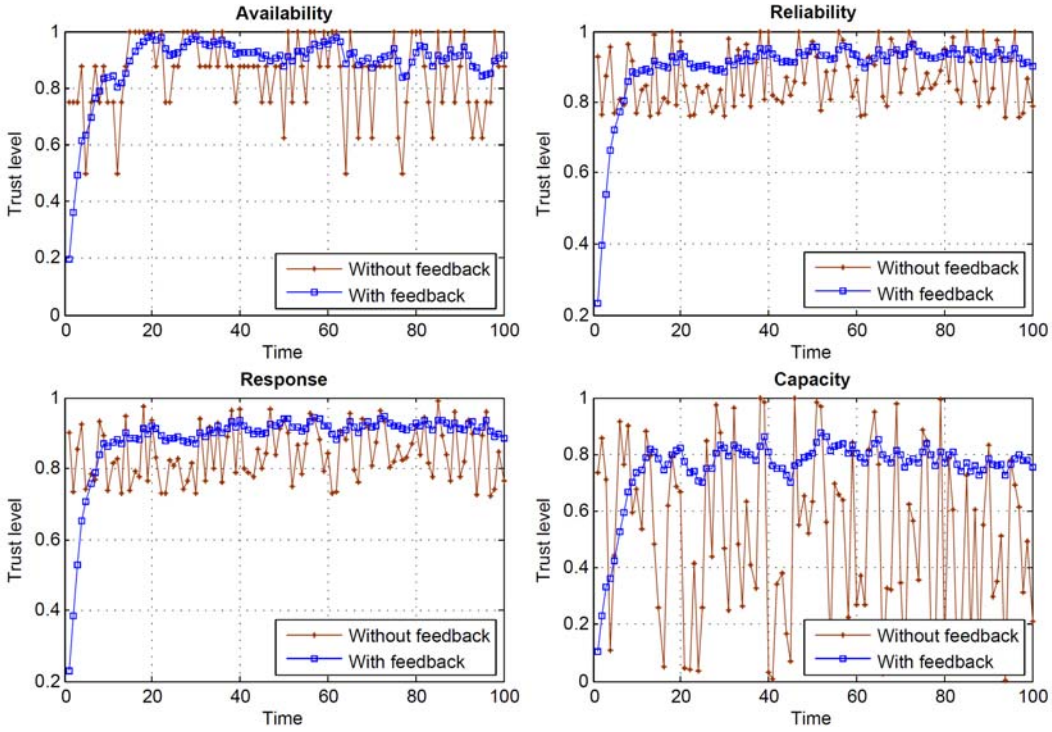

Figure **5**. The subplot describes level of trust against availability, reliability, response time and capacity.

made. Once, the results are integrated, the target environment still produces highly varying set of information which is not usable to make decisions. With our framework, we managed to normalize the results and let them fall within a short range which may alter based on the current state. Therefore, applied the framework the decisions made becomes trustworthy over the time. However, having a long history more accurate decisions could be made while reducing the seen dynamism when no feedback is applied.

8. CONCLUSION

Based on in-depth understanding of trust establishment process and quantitative comparison among trust establishment parameters, this paper presents an autonomic trust management framework for cloud based and highly dynamic IoT applications and services. We are increasingly aware of the necessity of eliminating the influence upon the evaluation results affected by malicious recommendation and defamation behaviors of a third party. In this framework, we adopt MAPE-K feedback control loop to evaluate the level of trust in an IoT cloud ecosystem. To evaluate the framework, we have developed a simulation framework. Thereby, we demonstrate consistency of the level of trust which is important with many IoT based dynamic applications and services. Apart from that, referring to the history of the records we enhance the level of trust at the same time. However, deployed the system, we expect it to improve further as history will accumulate over time. The trust management framework proposed for cloud based IoT system have been extensively studied with respect to their capability, availability, reliability and response time in practical heterogeneous cloud environment and their implementabilty. In the evaluation, it is evident that contributions from different parameters could be customized to fit into a specific context as it would be needed by a client. This enhances the flexibility of the system and let users to customize on their own need.

9. REFERENCES

[1] Luigi Atzori, Antonio Iera, and Giacomo Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] Fenye Bao and Ing-Ray Chen, "Dynamic trust management for internet of things applications," pp. 1–6, 2012.

[4] Matt Blaze, Joan Feigenbaum, and Jack Lacy, "Decentralized trust management," pp. 164–173, 1996.

[5] Riaz Ahmed Shaikh, Hassan Jameel, Brian J d'Auriol, Heejo Lee, Sungyoung Lee, and Young-Jae Song, "Group-based trust management scheme for clustered wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 11, pp. 1698–1712, 2009.

[6] Fenye Bao, Ing-Ray Chen, MoonJeong Chang, and Jin-Hee Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 169–183, 2012.

[7] Javier Lopez, Rodrigo Roman, Isaac Agudo, and Carmen Fernandez-Gago, "Trust management systems for wireless sensor networks: Best practices," *Computer Communications*, vol. 33, no. 9, pp. 1086–1093, 2010.

[8] Sheikh Mahbub Habib, Sebastian Ries, and Max Mühlhäuser, "Towards a trust management system for cloud computing," pp. 933–939, 2011.

[9] Hassan Takabi, James BD Joshi, and Gail-Joon Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security & Privacy*, , no. 6, pp. 24–31, 2010.

[10] Kai Hwang and Deyi Li, "Trusted cloud computing with secure resources and data coloring," *Internet Computing, IEEE*, vol. 14, no. 5, pp. 14–22, 2010.

[11] K.M. Khan and Q. Malluhi, "Establishing Trust in Cloud Computing," *IT Professional*, vol. 12, no. 5, pp. 20–27, Sept 2010.

[12] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[13] Siani Pearson and Azzedine Benameur, "Privacy, security and trust issues arising from cloud computing," pp. 693–702, 2010.

[14] Ryan KL Ko, Peter Jagadpramana, Miranda Mowbray, Siani Pearson, Markus Kirchberg, Qianhui Liang, and Bu Sung Lee, "TrustCloud: A framework for accountability and trust in cloud computing," pp. 584–588, 2011.

[15] Zheng Yan, Peng Zhang, and Athanasios V Vasilakos, "A survey on trust management for Internet of Things," *Journal of network and computer applications*, vol. 42, pp. 120–134, 2014.

[16] Paul Manuel, "A trust model of cloud computing based on Quality of Service," *Annals of Operations Research*, pp. 1–12, 2013.

[17] Dong Chen, Guiran Chang, Dawei Sun, Jiajia Li, Jie Jia, and Xingwei Wang, "Trm-iot: A trust management model based on fuzzy reputation for internet of things," *Computer Science and Information Systems*, vol. 8, no. 4, pp. 1207–1228, 2011.

[18] Mohamed Firdhous, Osman Ghazali, and Suhaidi Hassan, "Trust management in cloud computing: a critical review," *arXiv preprint arXiv:1211.3979*, 2012.

[19] Talal H Noor and Quan Z Sheng, "Trust as a service: a framework for trust management in cloud environments," pp. 314–321, 2011.