2011 IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing

A Learning Automata Based Solution for Preventing Distributed Denial of Service in Internet of Things

Sudip Misra
School of Information Tech
Indian Institute of Technology
Kharagpur, India
sudip_misra@yahoo.com

P. Venkata Krishna, Harshit Agarwal, Antriksh Saxena
School of Computing Science and Engineering
VIT University
Vellore, India
parimalavk@gmail.com, harshit.agarwal@live.com,
antrromet@gmail.com

Mohammad S. Obaidat,
Fellow of IEEE & Fellow of SCS
Dept. of Computer Science &
Software Engineering
Monmouth University, NJ, USA
obaidat@monmouth.edu

Abstract— **Internet of Things (IoT) refers to the networked interconnection of everyday objects. IoT is an upcoming research field and is being regarded as the revolution in the world of communication because of its extensible applications in numerous fields. Due to open and self-assimilation nature of these networks they are highly prone to attacks. Because of this reason security is of primary concern here. The security attack can be of various types, the idea here is to prevent IoT networks from Distributed Denial of Service (DDoS) attack. The objective of Denial of Service (DoS) is to make the server resources unavailable to the intended user, and when several such DoS attacks are present in a network then the attack is known as a DDoS attack. Our strategy is to prevent DDoS attack in IoT networks by using Learning Automata (LA) concepts. In this paper, we present a Service Oriented Architecture (SOA) which is used as a system model for IoT here. SOA provides a platform to the developers using which they can develop various applications for IoT without any concern regarding the nature of the objects, thereby acting as a middleware. The DDoS prevention strategy has been targeted for the SOA based architecture for IoT. The simulation results show that the proposed scheme is effective in preventing DDoS attacks in IoT.**

Keywords- Internet of things, distributed denial of service (DDoS), service oriented archtitecture (SOA), learning automata, cross-layer model.

I. INTRODUCTION

The internet of things (IoT) is an imminent model in the field of wireless telecommunications. It is also considered as a third wave of information technology after the Internet and mobile communication. Basically, IoT is a wireless interconnected network of variety of objects such as radio frequency identification (RFID) tags, sensors, actuators, mobile phones and other types of wireless devices. It has extensible application in the areas such as public security, infrastructure development, modern agriculture, environment protection, urban management, healthcare, enhanced learning, and business service, among others. The technologies of the Internet of Things can potentially cause the integration of production and service management and the integration of physical and digital world; one of the most sought after feature in any interactive device or service today. IoT is a self-configuring wireless network of sensors where the primary goal of establishing connection is to offer interconnectivity of various objects. Here, to implement this novel idea each device is equipped with some communication components and a unique addressing system so that each object can be uniquely identified. The concept of IoT was coined by the Auto-Id center of the Massachusetts Institute of Technology (MIT) in 1999 [1-14].

The IoT can act as the bridge between the real world and the online world. In the near future IoT is expected to become a collection of millions of devices forming a global information system. With the help of unique identification management, standardized interoperable communication protocols, and the integration of the objects with advanced technologies such as broadband mobile communication and cloud computing, these devices can cooperate with the other devices in the network and they can share information about their states. These capabilities endow the objects with intelligent self-decision making and self-management competencies, thus making them self-sufficient and autonomous [12].

We use the SOA based framework for the IoT which simplifies the access to the various services delivered by IoT. IoT renders a range of applications because of its potential users. For efficient realization of these applications, the SOA based middleware approach is the most apposite one. The main idea behind choosing the service oriented methodology for IoT is because of its potential for huge number of applications in our day to day life [13, 14].

The Idea of IoT is quite simple, but difficult to realize. The main technological challenges while implementing IoT is that all kinds of devices should be widely accepted thus providing interoperability between them. They should be capable of adapting to different networks during which their autonomous behavior should be upheld. Simultaneously the trust, security and privacy of the devices in the network should be preserved. These challenges can be encountered with the proposed cross-layer model which offers adaptability along with security.

Since in these networks, the new objects that enter a network are configured automatically, this accounts for the open nature of the IoT. This characteristic of the IoT makes it highly susceptible to security attacks. The objective of

978-0-7695-4580-6/11 $26.00 © 2011 IEEE
DOI 10.1109/iThings/CPSCom.2011.84

such attacks may be to simply disrupt the services provided by it or to gain administrative control over the network. There can be a variety of security attacks such as physical attack on the devices, eavesdropping, etc. The security issue being addressed here is the prevention of IOT networks from Distributed Denial of Service (DDoS) attack. The primary objective of the Denial of Service (DoS) attack is to flood the network with false service requests to the server, thus depriving the service to the legitimate requests. DoS is an attempt to exhaust the computational resources of the server, so as to make it unavailable to the intended user. When DoS attack is made by multiple agents in the network and from various locations then the attack is termed as DDoS. Each server has limited computational resources that are used for serving the requests and particularly it can handle only a limited volume of requests. During a DDoS attack, the server is inundated with requests. The server allocates the resources to each of these requests, but due to immense requests the server reaches its threshold capacity after which it cannot service other requests. Moreover, any request being serviced is most probably a fraud one. This prevents the server from providing service to the legitimate users [13].

In this paper, we use the concept of learning automata to devise a strategy for the prevention of the DDoS attack in the context of SOA for IoT. To counter DDoS at every level, we also suggest a cross-layer model for the same. Initially the DDoS attack in the IoT is identified, and then the necessary measures are taken to minimize the number of requests that are coming from the attacker.

II. MOTIVATION

As we have already discussed, IoTs are highly vulnerable to security attacks, there is an imperative requirement of proper strategies to be developed for countering such security threats. Since IoT is an emerging research field and is still under budding phase there is little work done for indemnification of communication in IoT. Especially, security attacks such as DDoS are almost left untouched. Because of heterogeneity in IoT it is extremely difficult to have homogenous communication standards being followed all over the networks. Different devices have different computational resources and capabilities. The standard should be such that it should offer interoperability and it should be compatible with all the nodes in the network. The counter-attack strategy should have low hardware requirements so that it can be implemented on all kinds of devices from the one which is low on system specification to the one which has abundant resources obviously. It is a very clumsy approach to develop different strategies for each device. Therefore, there is a need for an intelligent scheme which can adapt itself to different devices and at the same time can effectively prevent it from security threats. Since the main strength of IoT lies in the idea that it can support several sort of objects, consequently no compromises can be made on diversity.

DDoS is one of the serious issues as far as the security matters in the IoTs are concerned. Because of DDoS genuine service requests are not serviced. The open nature of IoT permits easy assimilation of attacker into the network. The attack temporarily brings down the serving device (server), as a result of which the other IoT objects are deprived of the services they deserve. During DDoS the network is flooded with counterfeit request which may lead to congestion in the network. This in turn may cause the energy dissipation of devices, which further adds to the problem as most of the objects are constrained by energy. Because of these costly consequences of DDoS attack, it has become extremely essential to develop a counter strategy which can prevent the attack as well as can identify the attacker(s) without wasting their energy needlessly.

The basic approach to prevent the attack in a network would be to sample the packets and identify the malicious one and drop them. Since sampling is purely an overhead on the network, the sampling of packets should be random and the sampling rate should be optimum to the situation. This is where LA comes into the picture; it intelligently determines the optimum sampling rate apt for the situation as well as for the object. It learns from the environment based on which it decides the action to be taken.

One more important characteristic of our approach is that it is not specific to any particular layer of network model. The strategy supports the Cross-Layer methodology and is applicable to all the layers in the network model. The cross layer approach enables the prevention of DDoS attacks at all the levels, hence making the scheme comprehensive in nature and thus more effective.

We have used this approach in context of the SOA model for IoT. Since, the primary purpose of IoT is to provide service to the intended users it is justified to use SOA tactic. The SOA models are gaining popularity in other fields such as Internet, and cloud computing, where the sole purpose is to lend services to users [14]. The LA based methodology in context of the SOA for IoT is a novel idea. We have not found any similar work which addresses the DDoS issue in IoT using LA based approach. Our approach is not parochial to any specific type of devices or domain. The use of LA and cross layer architecture helps in adapting to the different kind of objects, which is one of the most challenging problems in IoT.

III. LEARNING AUTOMATA (LA)

The theory of LA revolves around the notion of an

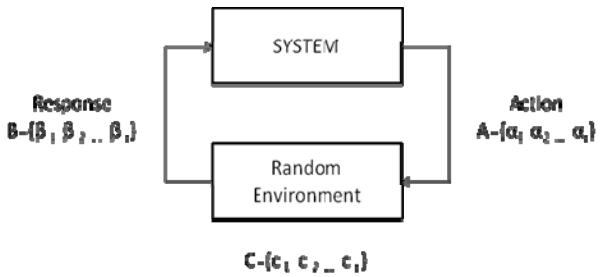

Figure 1. The learning automation

"*automaton,*" which is a self-operating machine or a mechanism that responds to a sequence of instructions in a certain way, so as to achieve a certain goal. The automaton either responds to a pre-determined set of rules, or adapts to the environmental dynamics in which it operates. The term "*learning*" refers to the action of procuring knowledge and modifying one's behavior based on the experience earned. Thus, the learning automata adapt to the responses from the environment through a series of interactions within them. The automata, then, attempt to learn the best action from a set of possible actions that are offered to them by the random stationary or non-stationary environment in which they operate. The automata, thus, act as decision makers to arrive at the best action.

LA can be used for optimization problems, since it selects that action which is more likely to be awarded by the environment. Over a period of time, LA learns from its actions and chooses an optimal solution. A comprehensive overview of LA can be found in the classic text by Narendra and Thathachar [5] and in the recent book chapter by Oommen and Misra [6].

A. The Automation

The automaton can be represented as a quintuple represented as $\{Q, A, B, F, H\}$, where [4]:

- Q is the finite set of internal states $Q = \{q_1, q_2, q_3 \ldots q_n\}$ where q_n is the state of the automaton at instant n.
- A is a finite set of actions performed by the automaton. $A = \{\alpha_1, \alpha_2 \ldots \alpha_n\}$ where α_n is the action performed by the automaton at instant n.
- B is a finite set of responses from the environment. $B = \{\beta_1, \beta_2, \beta_3 \ldots \beta_n\}$ where β_n is the response from the environment at an instant n.
- F is a mapping function. It maps the current state and input to the next state of the automaton; $Q \times B \rightarrow Q$.
- H is a mapping function. It maps the current state and response from the environment to determine the next action to be performed.

B. The Environment

The environment corresponds to the medium in which the automaton functions. Mathematically, an environment can be abstracted as a triple $\{A, B, C\}$. A, B, and C are defined as follows [4]:

- $A = \{\alpha_1, \alpha_1, \ldots, \alpha_n\}$ represents a finite input set;
- $B = \{\beta_1, \beta_2, \ldots, \beta_n\}$ is the output set of the environment; and
- $C = \{c_1, c_2, \ldots, c_n\}$ is a set of penalty probabilities, where element $c_i \in C$ corresponds to an input action α_i.

We now provide a few important definitions used in the field of LA. Given an action probability vector **P** (t) at time t, the *average penalty* is defined as [4]:

$$M(t) = E[\beta(t) \mid P(t)] = \Pr[\beta(t) = 1 \mid P(t)]$$

$$= \sum_{i=1}^{r} \Pr[\beta(t) = 1 \mid \alpha(t) = \alpha_i] \times \Pr[\alpha(t) = \alpha_i]$$

$$= \sum_{i=1}^{r} c_i p_i(t).$$

(1)

The average penalty for the "pure-chance" automation is given by [4]:

$$M_0 = \frac{1}{r} \sum_{i=1}^{r} c_i .$$

(2)

As $t \rightarrow \infty$ if the average penalty $M(t) < M_0$, at least asymptotically, the automaton is generally considered to be better than the pure-chance automaton. $E[M(t)]$ is given by [4]:

$$E[M(t)] = E\{E[\beta(t) \mid P(t)]\} = E[\beta(t)].$$

(3)

IV. SOA: OUR SYSTEM MODEL

The SOA (Service Oriented Architecture) model is a middleware approach for IoT (Internet of Things). Middleware is a layer or a set of layers which is placed between the actual technological infrastructure and the application level. The basic idea behind proposing a middleware for a system is that it extremely simplifies the development of the applications for the IoT infrastructure. It lets the programmers not to worry about the basic implementation of IoT, instead they just need to focus on accessing the services suitable for the application being developed by hiding complex details from them. SOA architecture is used for the IoT structure because it decomposes the complex applications into simpler monolithic components. The SOA based architectures lead to the fast and simplified development of the various business applications. Since SOA has already been proved a successful model for the Internet and is being used for providing various Internet based services today, it seems a good candidate for IoT. Moreover, it is worth mentioning that IoT is very much analogous to Internet but on a larger scale and having more heterogeneity [1,14].

In our SOA model which is quite similar to the one described in [1,11] consists of following layers:

A. Applications

This is the topmost layer of the architecture. It provides all the system services to the end user. This layer is not a part

of middleware, instead it just accesses the middleware so as to provide the functionalities of the system to the user.

B. *Service Collection*

This layer is a common layer for the SOA-based middleware architecture. It provides all the functionalities of the system to the layers above it. It is an abstract layer where all the infrastructure of the system is hidden and only the services are revealed. In other words, it is a repository of the services that the underlying technological buildup delivers. The reason behind the creation of this layer is that the application developer need not be concerned with the internal working of the systems and the simple representation of complex processes which makes it easier for the designer to understand and design the application.

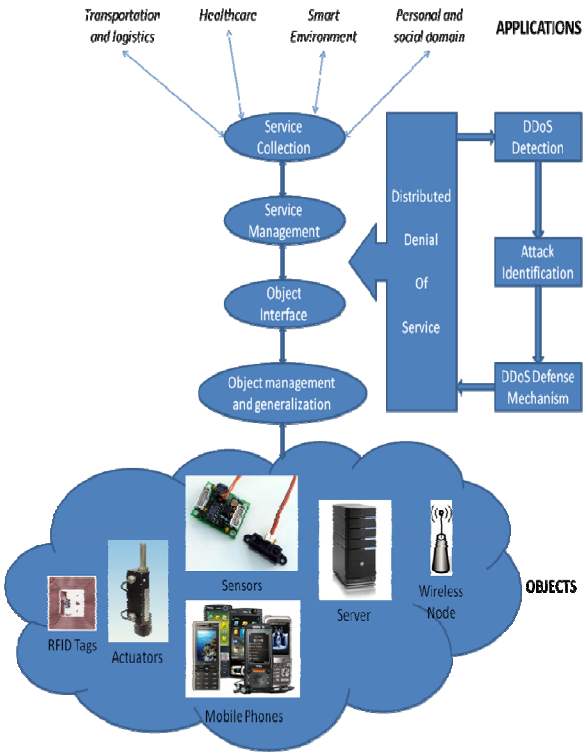

Figure 2. Service Oriented Architecture for IoT

C. *Service Management*

All the services management issues such as service status monitoring, service queuing and service configuration are handled at this layer. This layer also consists of a central repository which contains the details regarding which set of objects or object is associated with a particular service. The service management in this layer is application dependent and is adapted to applications in such a way that we get the optimized result for every kind of application.

D. *Object Interface*

This layer provides access to all the functionalities of the objects. This is an abstract layer which hides the objects but show only their functionalities to the above layer. This layer acts as an interface between the objects and the service management layer. With the introduction of this layer it is easy for the service management layer to access the functions of the various objects without going into their details. Rather than worrying about how to implement the functions, the upper layers just needs to specify the functions that it wants to access.

E. *Object Management and Generalization*

IoT is a collection of heterogeneous nodes which can consists of various objects such as RFID tags, actuators, sensors, servers, wireless devices, etc. Since the objects are heterogeneous, this layer provides an interconnection between them with the help of some standardized protocols. It communicates with all objects in a way that is compatible with them. This helps in increasing the homogeneity and promotes harmony among the objects. With all these characteristics, it makes it easier for the upper layers to have access to various objects. The upper layers need not worry about the type of the object it is dealing with rather it just needs to give the generalized set of commands required to perform a particular function. Hence, in this way this layer generalizes all the heterogeneous objects as a homogeneous one.

The layer also performs various object management related tasks such as self-organization, self-configuration and self-healing of the objects.

As this layer deals with the various objects directly, it is most susceptible to various security attacks such as DDoS hence; the best suited to implement our counter DDoS strategy. We propose the counter DDoS strategy which uses the concepts of Learning Automata (LA) to prevent the DDoS attack in an efficient manner. Since the devices in IoT are constrained by the computational resources, we make this process an efficient one so that in minimum amount of energy being consumed and latency introduced, we associate a self-learning mechanism (LA) with each object in IoT.

F. *Objects*

This is the bottom most layer of the architecture. It consists of all the objects that are present in the IoT such as RFID tags, sensors, actuators, servers, wireless devices, etc. This layer is not a part of the middleware though it forms a network of IoT. It is the basic infrastructure which is required to run any IoT based application. IoT consists of various day to day objects, hence these objects are constrained by the energy and computational power. These restrictions are so dominant that while designing any communication protocol for the devices these constraints are the primary concern.

From the broader perspective SOA can be viewed as analogous to the cross-layer network model. The layers such as the applications, service collection, service management and object interface together forms the application layer. The transport and network layer come into being in object management and generalization layer of the SOA. Remaining last two layers are the MAC and physical layer, which are members of the lowermost layer of the SOA, i.e.

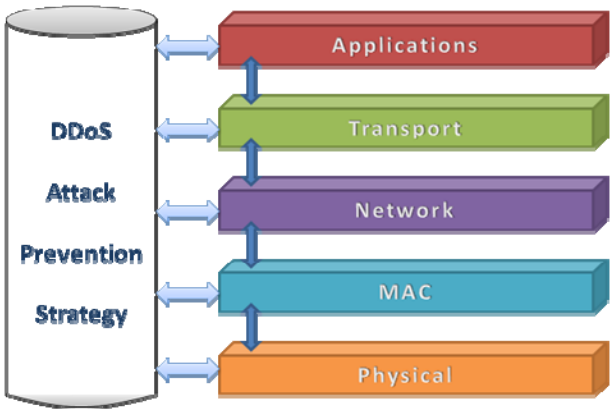

Figure 3. Crosslayer network model for IoT objects

objects layer. The DDoS prevention strategy works over all these layers to preclude them from any possible DDoS attack. All these layers and DDoS prevention component works in unison with each other, thus carving out a cross-layer model. This model is explained in detail in the next section.

V. CROSS-LAYER NETWORK MODEL

In this section we enlarge the approach we used for the prevention of DDoS attacks in IoT. The approach is subdivided into two major components, first the strategy which was used to counter DDoS and second the set of layers of network model and their linkage to the DDoS prevention strategy component. The scope of the proposed strategy is not limited to any specific function layer in a device. The cross-layer based proposals, especially for the wireless networks, are quite popular, primarily because the conventional layered model does not sufficiently serve all the needs of wireless mode of communication. Many researchers like in [8,9] have made use of this model to either gain performance or provide more services in wireless networks. The conventional network layer model was scrutinized primarily because of two reasons. Firstly, the DDoS attack is not restricted to any particular layer and can take place at any one of them, and secondly, it does not support the kind of heterogeneity characteristic of IoT. Therefore, we use cross-layer model to ensure that every layer is prevented from the possible DDoS attack. As shown in Figure 2, each level interacts with the DDoS prevention strategy component and vice versa. The model is referred to as cross-layer because every layer interacts with the common component while maintaining their characteristic nature [10].

A. DDoS attack prevention strategy

Before we explain our solution, we first introduce the LA concepts which are the crux of our strategy. The whole idea was inspired by reference [3], which gave a LA based solution to the DDoS attacks in wireless mesh networks (WMN).

1) The LA system model

The LA-based model for DDoS prevention in the context of IoT is given by following parameters.

- $\alpha\{\alpha_1,\alpha_2,\ldots,\alpha_r\}$ is the set of sampling rates in the system;
- β is the environment response for an action α_n.

The automation is located inside the DDoS prevention strategy component whereas network model layers act as the random environment. Based on the response given by the environment for an action α_i the next best suited action from the given input action set α is selected.

2) Phases

As given in [3], our approach also consists of three phase DDoS detection, identification and defense.

a) DDoS Detection

For handling any request of the client, the server needs first to allocate resources to the client. Similarly in the context of network model layers, each layer is capable of handling some limited amount of data or request in a given interval of time. Whenever the rate of incoming traffic exceeds the threshold DoS occurs.

To detect a DDoS attack, at each layer a maximum servicing capacity or threshold is defined. This limit is defined as per the computational resources availability. Therefore, its value varies with different layers and different objects. In this phase, the DDoS prevention component in each device monitors the number of requests that each layer is receiving. If the number of requests exceeds the threshold value of any of the layer, then a DALERT (DDoS alert) is issued by the component. This DALERT is transmitted to all the immediate neighbors of the object, which in turn further propagate this alert to other nodes. The issuing of the DALERT embarks the starting of the next phase of DDoS prevention strategy, i.e. attack identification phase [3].

In nutshell, this phase detects a DDoS attack and is carried out entirely at the servicing device i.e. server. During this phase other objects keep functioning in the regular manner. The DDoS detection does not confirm an attack, but it merely indicates the possibility of one.

b) Attack Identification

After receiving the DALERT message devices are conscious that there is possible DoS attack on server in the IoT. To identify the device that is attacking, the device starts sampling the data it is transmitting and will make list of all the hosts that are sending the request. The device that is attacking will obviously be sending more number of requests as compared to other hosts. The ID of the device which is sending the largest number of requests will be determined.

This determined identity of the attacker is passed to all the devices in the network, so that they are alerted and go into DDoS defense phase. The attacker's information is sent using a special packet known as Attacker Information Packet (AIP). The AIP contains IDs of all the hosts which

are trying to attack the server. Upon receiving this packet all the nodes will go into the DDoS defense mode [3].

c) DDoS Defence

Once the attacker has been identified, and all the nodes are informed of this, the nodes start sampling the incoming traffic. The packet which is identified as coming from the attackers is discarded. It is this phase which makes use of LA concepts to determine the sampling rate.

Scanning and sampling all the packets that go through the system is not a feasible approach as it may lead to latency in the system, thus making the system more vulnerable to attacks. A cost is linked with the sampling of each packet. Additionally each object is assigned with some sampling budget based on its computational capabilities and energy. The maximum rate at which a node can sample data is called sampling budget for that node. The budget is assigned such that if sampling is performed at this rate then it will result in acceptable latency in the network. One simple solution to the problem could be sampling the maximum number of packets possible with the allocated budget, thus maximizing the chances of malicious packets to

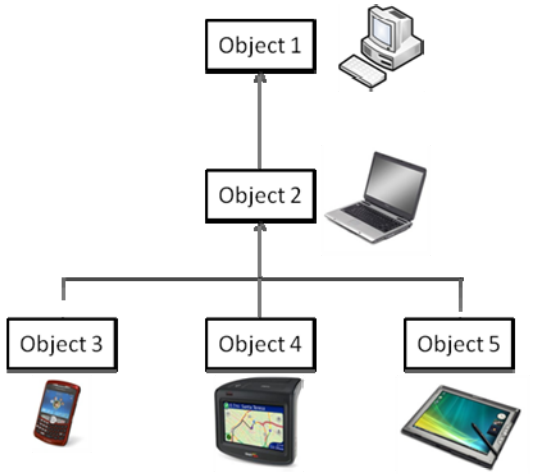

Figure 4. Simulation Model

be dropped. However, the above solution is not efficient, as the genuine packets will be sampled needlessly. In order to make the process of sampling an efficient one so that in minimum amount of energy being consumed and latency introduced, maximum number of malicious packets is sampled, a self-learning mechanism is associated with each node in IoT [7].

In essence, DDoS defense phase makes use of LA to establish the optimum sampling rate. During sampling as the request sent by attacker is discarded, obviously the number of requests at the server would reduce and the IoT would be free of the DDoS attack. But this mechanism does not guarantee the dismissal of all the malicious packets. As only few selected packets are sampled some malicious packets will be transmitted. The sampling process is carried out at each node once the DDoS defense phase starts therefore the malicious packet which was left undetected at a node might be dropped at the other one. This will increase the path followed by that packet, which can boost its chance of being detected [3]. The end of this phase is reached when the number of service requests to the server drops down below the server's servicing threshold. The server then issues an alert to all other nodes informing them about the successful aversion of the attack. Upon receiving the alert all the nodes start functioning in their regular manner.

B. Network stack

This component of the cross-layer model comprises of set of logical layers of network stack and their association with the DDoS prevention strategy component. As already explained above, these layers correspond to different layers of SOA model. Each layer has some functional characteristics associated with it. These functions are logically clustered on the grounds of task performed by them, into various layers. For instance the applications, service collection, service management and object interface layers of SOA, perform application level jobs, hence they can be mapped into the application layer of network stack.

In our case, every layer continues to function typically as per the convention defined by various protocols and standards, except that the DDoS prevention component observes for any possibility of DDoS attack and averts if there is any. As discussed above, the DDoS attack takes place when the device receives more number of requests than its capacity or threshold. Therefore, for each layer there is a threshold value which defines its maximum capacity to service per unit time. The DDoS prevention strategy component is informed of this maximum limit of each layer, so that it can detect a DDoS attack on them. The network

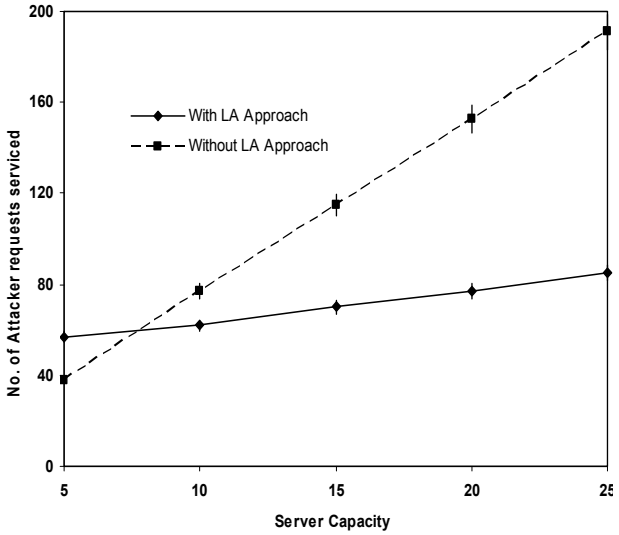

Figure 5. Service attacker request versus server capacity

stack is also responsible for sending any DDoS prevention scheme related data it receives such as DALERT or AIP to the DDoS prevention component.

In context of LA model, the network stack functions as a random environment, and its response is given to the automation placed inside the DDoS prevention strategy component.

VI. SIMULATION

The performance of the proposed strategy was evaluated by simulating under the IoT circumstances. These IoT devices or objects possessed various features such as self-organization, self-configuration, heterogeneity etc., which are characteristics of IoT, hence emulating an IoT like environment. The simulation results suggest that our DDoS attack prevention strategy is effective and applicable in IoT context. In the following section we will explain simulation model and give an analysis of its outcome.

A. Setting and configuration

The test bed for our simulation model consists of five objects. One object was connected to Internet and has the capability of providing Internet connectivity to other objects and remaining devices were just accessing its services or performing the routing. As depicted in Figure 4, object 2 was behaving like a simple bridge in the network, which was forwarding the request on behalf of other objects to the serving object. We have assumed that the object 2 was not requesting any services and was just performing the task of forwarding the requests it received from the other objects. Object 1 was acting like an Internet provider which was solely meant for serving the request of other devices. Objects 3, 4, and 5 were acting as fundamental devices whose only job was to request the server for Internet service. Since object 2 was the closest device in their proximity, they were sending their requests to it, which consequently was forwarding their requests to the server.

As discussed above, our approach is based on cross-layer model and is capable of handling the attack at any level of network stack. In this case, the strategy was evaluated by launching a DDoS attack on the application layer of the serving device. To make the attack distributed in nature, it was carried out simultaneously from object 3 and 4. Therefore, we have two attackers and one legitimate user; object 5. A summary of all the experimental parameters is given in Table1.

Table 1. Simulation parameters

Parameters	***Value***
Simulation time	100s
LA reward parameter	0.1
LA penalization parameter	0.6
Server threshold	80%
No. of service request sent by attackers	83
Rate of attack by object 3	50
Rate of attack by object 4	33

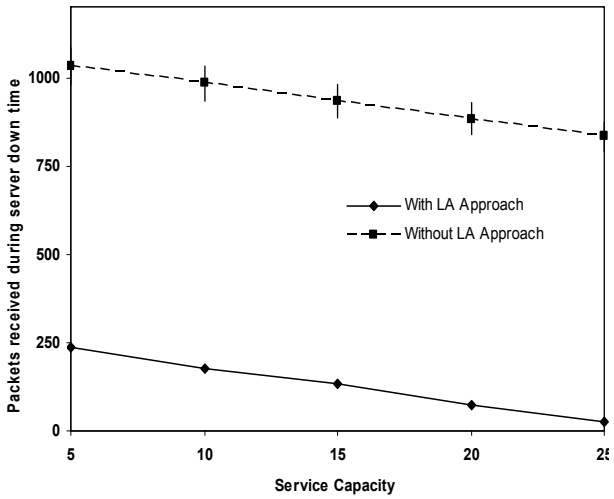

Figure 6. Denied Service Capacity

B. Results

Figure 5 shows the variation of packets denied from being serviced during object 1's down time with its servicing capacity. As evident from the Figure, we see that the number of requests denied when LA was implemented was far less than the value in LA's absence. Since the number of malicious service requests was dropped considerably by LA, thereby the total requests received by the server also declined significantly. The graph shows increase in the number of packets serviced as the server capacity increases. This result is obvious for both the situations, as the capacity of the server to handle the requests increases and object 1 will be able to provide services to more number of packets whether they are malicious or not. The experiment was carried out to study the performance of the strategy when the server was overwhelmed with servicing requests.

In Figure 6, a graph of the number of requests of attackers serviced versus server capacity has been plotted for both with and without LA implementation. This includes the requests serviced both during and before its down time. When LA is operational the number of attackers' requests was quite less as compared to when it was absent. Because in the case of our strategy, during sampling many attacker packets were dropped and thus the server received lesser number of attackers' requests. As the servicing capacity increases the object 1 was able to handle more number of requests, as a result of which more requests from the attacker were serviced with the increase in the servicing capacity. And the same result is manifested from the graph.

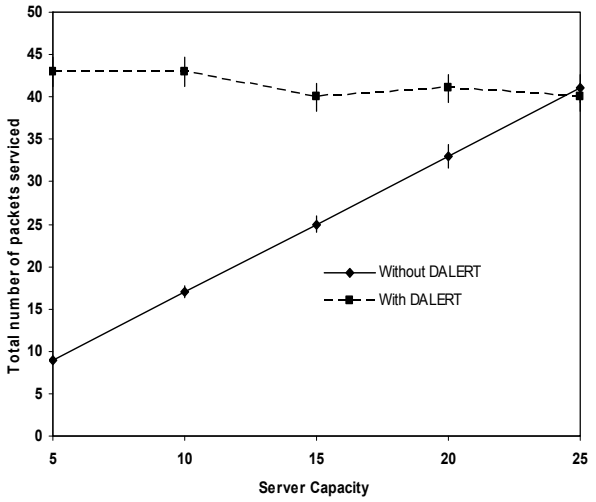

Figure 7. Packets serviced before and after DDoS detection

In order to illustrate the credibility of the proposed DDoS counter strategy, we analyzed the total packets serviced before and after the DDoS detection. The requests that are being sent to the server include both the attackers' request as well as the legitimate users' requests. The graph shown in Figure 7 is considered to be vital because it exhibits the drop in the number of total serviced requests by the server after the DDoS attack detection. According to our strategy, once the DDoS attack has been identified, the sampling process starts during which a number of packets coming from the hosts identified as attackers are dropped. Before the DDoS detection there is a gradual increase in the total requests serviced with the increasing server capacity because the increase in server capacity delays the occurrence of DDoS due to which the server threshold is reached later in time. The rate of sending the requests to the server by all the objects remains constant. After the

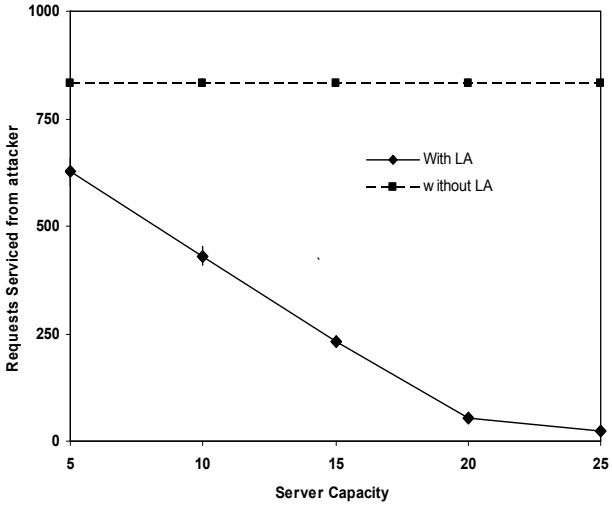

Figure 7. Packet dropping behavior of the IoT objects

detection of DDoS, at each server capacity the same number of malicious requests was dropped and also the same legitimate requests were served due to which the total packets being serviced remained constant. In essence, the graph implies that after the DDoS attack discovery the number of packets of the attackers serviced are reduced considerably, as a result of which the attack on the server is averted and it continues to render uninterrupted services to the legitimate users.

The graph in Figure 8 depicts the packet dropping behavior of the nodes with the changing sampling budget, i.e. the maximum rate at which the node can sample data. A comparative study between the results obtained with and without the LA implementation has been done. It is clear from the results that as the sampling budget increases the ability of LA to drop malicious packets improve. Consequently, the number of service requests by attackers received by server decreases drastically. When LA is not implemented, there is no logic that checks for the malicious requests and hence the total requests from the attacker remain constant in this case.

VII. CONCLUSION

This paper presents a LA-based preventive scheme for DDoS attack for IoT. Looking at the ability of IoT for use in a variety of application domains, we employ service oriented architecture to harness the full potential out of IoT. The SOA model not only provides services of IoT, it also ensures that the whole framework is free from the DDoS attack. Along with SOA, in order to explain the strategy in context with the network layer model, we introduce the concept of cross-layer model so that the strategy to prevent the DDoS attack could be applied at every layer of the network stack, thereby making it compatible with different types of objects as well.

It can be inferred from above that LA acts as a pivot of our DDoS prevention strategy. Knowing the critical nature of parameters such as energy and computational power in IoT objects, the LA scheme effectively prevents the objects from a DDoS attack by making the optimal utilization of its resources. The LA component of the strategy attributes to its adaptive nature to various situations, thus making it suitable for diverse network settings and operating conditions. The efficacy of our strategy is validated by the experiment results.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey, " Computer Networks, Vol. 54, No. 15, pp. 2787-2805, October 2010.

[2] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "IOT Gateway: BridgingWireless Sensor Networks into Internet of Things," *Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on* , Vol., No., pp.347-352, 11-13 Dec. 2010.

[3] S. Misra, P. Venkata Krishna, K. I. Abraham, N. Sasikumar, and S. Fredun, "An Adaptive Learning Routing Protocol for the Prevention of Distributed

Denial of Service Attacks in Wireless Mesh Networks," Computers & Mathematics with Applications, Vol. 60, No. 2, pp. 294-306, in Advances in Cryptography, Security and Applications for Future Computer Science, July 2010.

[4] M. Esnaashari, M. R. Meybodi, "Data Aggregation in Sensor Networks Using Learning Automata," Wireless Networks, Vol. 16, No. 3, pp. 687 – 699, 2010.

[5] K.S. Narendra, M.A.L. Thathachar, "Learning Automata," Prentice-Hall, 1989.

[6] B. J. Oommen, S. Misra, "Cybernetics and Learning Automata, in Handbook of Automation (S. Nof (Ed.), Springer, 2009.

[7] S. Misra, P. V. Krishna, and K. I. Abraham," Adaptive Link-State Routing and Intrusion Detection in Wireless Mesh Networks," *Information Security, IET* , Vol.4, No.4, pp.374-389, December 2010.

[8] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-layer Design for Wireless Networks," *Communications Magazine, IEEE* , Vol.41, No.10, pp. 74- 80, Oct 2003.

[9] V. T. Raisinghani, and S. Iyer, "Cross-layer Feedback Architecture for Mobile Device Protocol Stacks," *Communications Magazine, IEEE* , Vol.44, No.1, pp. 85- 92, Jan. 2006.

[10] M. Srivastava, and M. Motani, "Cross-layer Design: A Survey and the Road Ahead," *Communications Magazine, IEEE* , Vol.43, No.12, pp. 112- 119, Dec. 2005.

[11] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. Souza, and V. Trifa, "SOA-Based Integration of the Internet of Things in Enterprise Services," Proeedinsg of the 2009 IEEE Intenrtaional Confernece on *Web Services,* pp. 968-975, July 2009.

[12] ITU Internet Reports, The Internet of Things, November 2005.

[13] J. Mirkovic and P. Reiher," A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," *SIGCOMM Comput. Commun. Rev., Vol. 34, No. 2, pp. 39-53, April 2004*.

[14] P. Ferreira, R. Martinho, and D. Domingos," IoT-aware Business Processes for Logistics: Limitations of Current Approaches," Proceeidngs of *InForum 201, pp. 611-622, Sep.2010.*