Journal of Universal Computer Science, vol. 26, no. 11 (2020), 1435-1454
submitted: 31/8/20, accepted: 2/10/20, appeared: 28/11/20 CC BY-ND 4.0

Fuzzy Adaptive Data Packets Control Algorithm for IoT System Protection

Łukasz Apiecionek
(Kazimierz Wielki University, Bydgoszcz, Kujawsko-Pomorskie, Poland
lukasz.apiecionek@ukw.edu.pl)

Matusz Biedziak
(Kazimierz Wielki University, Bydgoszcz, Kujawsko-Pomorskie, Poland
biedziak.m@gmail.com)

Abstract: One of huge problem for recent IT systems are attacks on their resources called Distributed Denial of Service attacks. Many servers which are accessible from public network were a victim of such attacks or could be in the future. Unfortunately, there is still no effective method for protecting network servers against source of the attack, while such attack could block network resources for many hours. Existing solutions for protecting networks and IoT systems are using mainly firewalls and IDS/IPS mechanisms, which is not sufficient. This article presents the method minimizing the DDoS attacks. Proposed method provides possibilities for the network administrators to protect their servers and IoT network resources during the attack. The proposed fuzzy adaptive algorithm is using Ordered Fuzzy Numbers for predicting amount of packets which could be passed over the network boarder gateway. Proposed solution will give the opportunity for ordinary users to finish their work when the attack occurs.

Keywords: IoT, DDoS, network security, critical infrastructure protection
Categories: C.2.1, D.4.6, F.2.1, K.6.5

1 Introduction

Nowadays everybody needs a fast access to information from every part of the network. The systems have to be accessible online all the time, while Distributed Denial of Service (DDoS in short) attacks have become a huge problem as they cause network unavailability by blocking services. This is made because such services seizing system resources in computers in the network until they stop working. Administrators are afraid of DDoS attacks on his systems [Mazur, 2019], especially the administrators responsible for Critical Infrastructure Protection (CIP) [Pietkiewicz, 2018].

Usually Critical Infrastructure refers to systems that have distributed natures, are comprised of physical elements that work in a real-time and are capable of communicating with each other [Choras, 16]. Such systems integrate computational, communication and physical aspects in order to improve usability, efficiency, and reliability. Unfortunately, this combinations introduce a wide spectrum of risks related to cyber domain. If we consider critical infrastructure as a systems for control of energy, water, and transportation as well as in the area of smart houses, autonomic vehicles in future, we could imagine the results of accident in such environment

[Andrysiak, 2020] [Bujnowski, 2020A] [Bujnowski, 2020B]. There is also an idea of Internet of Things (IoT in short), which consists in connecting all possible devices to the Internet in order to provide them with new functionalities and in this way – to improve the user's life standard [Sudmaeker, 10]. Figure 1 illustrates the overall concept of the IoT. In this idea every domain specific application is interacting with domain independent services, whereas in each domain, sensors and actuators communicate directly with each other [Al-Fuqaha, 15]. Some of the domain represent the Critical Infrastructure, e.g. the Intelligent Transport System (ITS in short) [Ambak, 09], healthcare, and industry. There are still some work in progress on ITS. This system in future should allow ensure green traffic light for emergency vehicles. The new feature in the future will be possibilities for preventing congestions in intelligent way, for example by manipulating the duration of the green light period for various traffic directions. But blocking this system could generate red light in whole city which cause huge traffic jams.

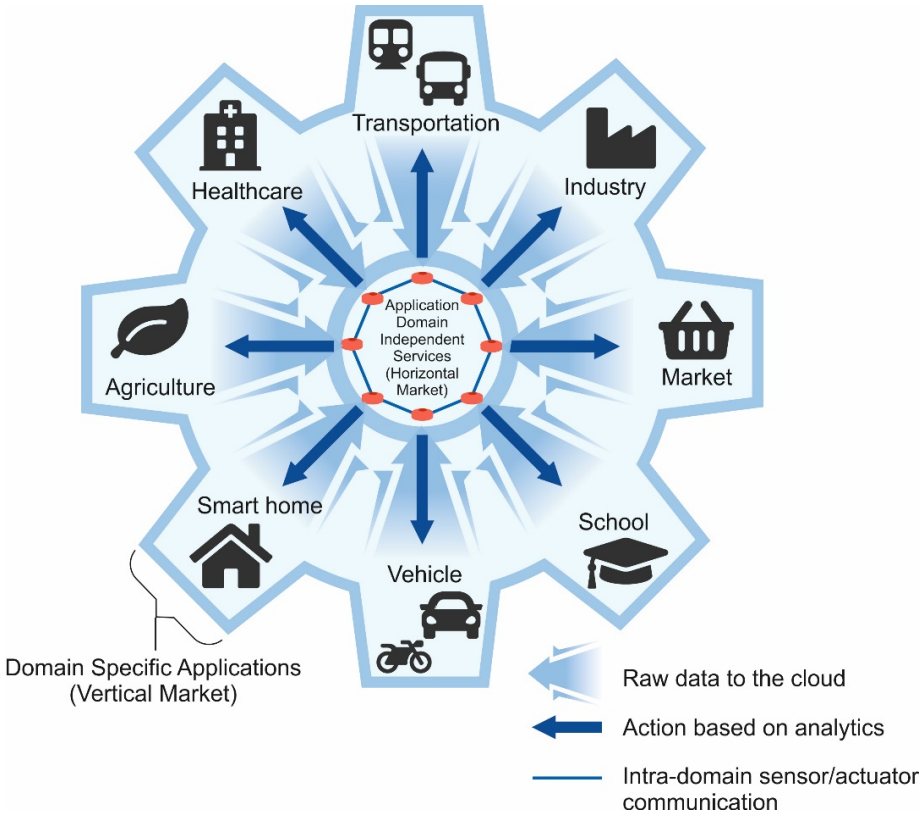

Figure 1: The overall picture of IoT emphasizing the vertical markets and the horizontal integration between them [Moor, 06].

The IoT systems could have the following types of architectures can be specified:

- Sensors,
- Fog,
- Cloud Computing.

The solutions are illustrated in figure 2. On the lowest level there are sensors. Senors are the IoT devices which are responsible for collecting data. They generate the highest requirements for the address pools and the network traffic when transmitting the acquired data due to their great number. The upper layer is the Fog-type solution. This layer gathers, aggregates and preliminarily processes the data collected by the lower layer. The solutions at this level are more complicated than at the sensor level. The upmost layer is Cloud. In this type of solution all data are processed in the cloud. It requires a suitable structure, building a centre for data processing and managing it in a proper manner, but also sending all the data to the center, where their processing takes place.

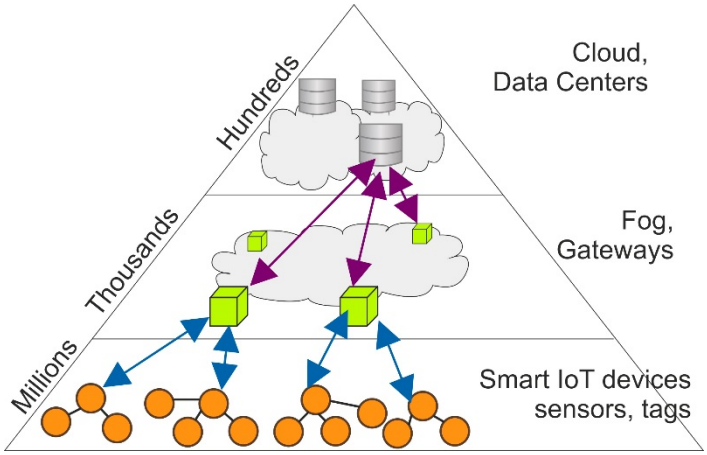

Figure 2: The IoT elements [Al-Fuqaha, 15].

The IoT system has a distributed architecture, so the analysis of the safety issues in those systems reveals the main challenges in this field:

- Data encryption,
- Availability,
- Confidentiality,
- Serviceability,
- Unique identification,
- Denial of Service attack possibility.

The IoT solutions are used mainly for monitoring physical parameters as well as making faster and more appropriate decisions. For instance, collecting sensitive information, such as the patients' health by means of remote monitoring, is restricted by law and in order to ensure their confidentiality they must be encrypted. Another

crucial aspect is availability. On choosing the IoT solution for example, to optimize the production process in a company, an access to the data which serve as the basis for the decisions must be provided. Lack of access to those information or unavailability of the device may be caused by the Denial of Service-type attack, which is able to block the access to the sensors. In such situation the industry system may stop the production line. Stopping this line may result in financial loss of the company. IoT systems as Critical Infrastructure security issues are complex and time-consuming, often requiring implementing new approaches. These approaches should require low computer power because only such method could be widely implemented in small IoT sensors.

Lot of Internet users could remember the situation, when they are started some work on the network and loses the connection unexpectedly. In such situation they cannot even log out of the system. There are still not any method which provide the solution for such problem. Usually papers present methods of fighting the DDoS attack problems [Rocky, 02] [Moor, 06] [Schuba, 96][CERT, 00] [CERT, 97] [Apiecionek, 14] [Renk, 11] by using the Intrusion Detection System and Intrusion Prevention System (IDS/IPS in short) solutions. Such systems are efficient provided that they have a description of well know attacks. In some solutions IDS/IPS are using some kind of Artificial Intelligence (AI in short). Using AI this systems could learn the actions in some specific scenarios of attack. Other solutions suggest using a firewall mounted on the network boarder. However, this firewall will only block the incoming traffic on specific ports or IP address ranges, which is not sufficient. There are some solutions which are sniffing data transfer, process this data and classified them. Then this information is used for blocking special kind of traffic on firewall. The D'Antonio et al. [D'Antonio, 06] presents such solutions with data mining process, which extracts behavioural models from pre-elaborated network traffic. Monitoring network traffic entails [D'Antonio, 06]:

- necessity for capturing packets from an observation point, e.g. an end-system, an access network link, or a backbone link,
- classifying observed packets in application-based meaningful sets,
- evaluating appropriate parameters from which user's behaviour can be inferred.

Ten et al. in [Ten, 10] proposed security SCADA framework. This framework encompasses four key components: real-time monitoring, anomaly detection, impact analysis, and mitigation strategies. Osanaiye et al. in [Osanaiye, 16] presents the cloud DDoS defenses possibilities. Such defenses can be deployed in four key locations: source-end, access point, intermediate network, and distributed. It also presents the client control solution architecture which use tokens for packet transfer though the network. If the token is invalid, the packet is blocked. While the Knowles et al. in [Knowles, 15] described such idea as a future research areas for industrial control system. Bernieri et al. in [Bernieri, 17] outline how the presence of a cyber Intrusion Detection System improves the effectiveness and the reliability of the protection schema. This paper also show how the systems works under cyberattacks. The DDoS protection method is presented by Kuilay et al. [Kubilay, 18]. Their method is using shuffing-based containment mechanism in order to quarantine malicious client.

In this paper authors presents a concept of new method. This method could be implemented in all possible places of the networks, because it not require huge computer power. This lets to implement method on firewalls and routers in the network. Using method in this equipment could eliminate the effects of DDoS attacks. The idea of developing this mechanism was to prepare method for dealing with DDoS attacks.

The structure of this paper is as follows. Section 2 the proposed method, provides the test results. Finally section 3 provides a conclusion and discussion over the developed method.

2 The idea of proposed method

The DDoS attacks were presented in many papers [Rocky, 02] [Moor, 06]. These attacks could be used on different system resources, e.g. on DNS servers or TCP/IP sockets [Moor, 06] [Schuba, 96]. The main principle of most methods is to start huge numbers of user connections. This number has to exceeds the possibility of the system for resource allocation. Then, such server status cause their abnormal work. Methods for dealing with the DDoS attacks by their global detection and the necessity of cooperation between network providers are described in many papers [Rocky, 02] [Moor, 06] [Schuba, 96] [CERT, 00][CERT, 97] [Apiecionek, 14]. But the facts is that the transmission of the attackers' packets is done through the provider's network. So, the packets transmitted through the network could provide to data link saturation. Then, the link saturation lead to lack of connection to the server. Unfortunately there is no communications between the aim of the attack and the provider network resources in the existing protection methods. The most common concern is the limited performance of network devices. However, the authors claim, that there is possibility to limit the incoming traffic on a firewall and then allow the servers to deal with the already established connection by the users. In proposed method, the server which is an aim of the attack, could provide the information to network provider, that it become the aim of the attack. This information let the provider's network to react in the attack. It could block special traffic to the server on their equipment. This should let the users finish their work and the new users also should be able to connect to the server.

The proposed method could be implemented on two main network elements:

- the server by adding the special module for their protection,
- the provider's network equipment: especially routers and firewalls, which could block specific traffic which represents the DDoS attacks packets.

2.1 Creation of Meta data

Most routers could block specific traffic. For this purpose there are access list mechanism which decide if the packet will be transferred to the network or not. This mechanism is also implemented on the firewalls. There were also some QoS method ideas which could work on one routers and try to protect network resources locally [Apiecionek, 14]. But such proposition will not provide any mechanism for recognition the source of the attack. That is why it will not solve the problem. The

hacker could still send their packet to the server and allocate servers resources. Lot of routers are implementing IPSec encryption with Public Key Infrastructure for other side authentication. This mechanism is very useful because could be implemented on server side to create secure connection between server and network equipment. Lot of public servers which have to be protected belong to banks. Banks are using mainly PKI and X.509 certificates for their authentication. This mechanism could be used in proposed servers protecting methods. Also, there are lot of methods for spreading information between routers like routing protocols [Piechowiak, 12A] [Piechowiak, 12B] or Simple Network Management Protocol. But there is no idea for whole solutions which could work on routers, firewalls, and servers, which will let to block the attackers traffic when DDoS attack will be recognized. That is way the authors proposed and develop one.

2.2 The algorithm used for the experiment

The algorithm used for the experiment was taken from Apiecionek et al. 2016 [Apiecionek, 16]. In this method the aim of the attack is a server. The server is the best place to recognize that the attack has been started. When the routers get the information, that they have to block some specific traffic, or traffic for specific server, they could use the mentioned mechanism already implemented on them - access lists. The first thing is the information process which pass the information from the server to the router that there is an attack. This information process is protected and the both side of the communication are authenticated. It is made using IPSec encryption and PKI. Mentioned IPsec communication channel is used for passing the two main information from the server. The first information is that the server is under attack. The second one is the information what kind of traffic has to be blocked in the whole network. In this channel a Simple Network Management Protocol is used as a protocol which pass the information very fast. SNMP is widely implemented on existing equipment: routers and firewalls. The method could works on servers, routers, and firewalls and has got a following steps:

- Step (1) - server is collecting information about their traffic statistic,
- Step (2) - server is recognizing that there are under attack using their traffic statistic and their work status e.g. connection number, RAM usage, processor usage,
- Step (3) - server is establishing IPSec channel to network provider's equipment: router or firewall,
- Step (4) - server is passing two information: it is under attack and what kind of traffic which has to be blocked by the network; for this purpose a SNMP over IPSec channel is used,
- Step (5) - router or firewall start to block specific traffic,
- Step (6) - router or firewall is spreading the information about specific traffic which has to be blocked via SNMP trap message to other routers and firewalls.
- Step (7) – the specific traffic of the attacker is blocked via the mentioned network devices.

These steps are presented in figure 3.

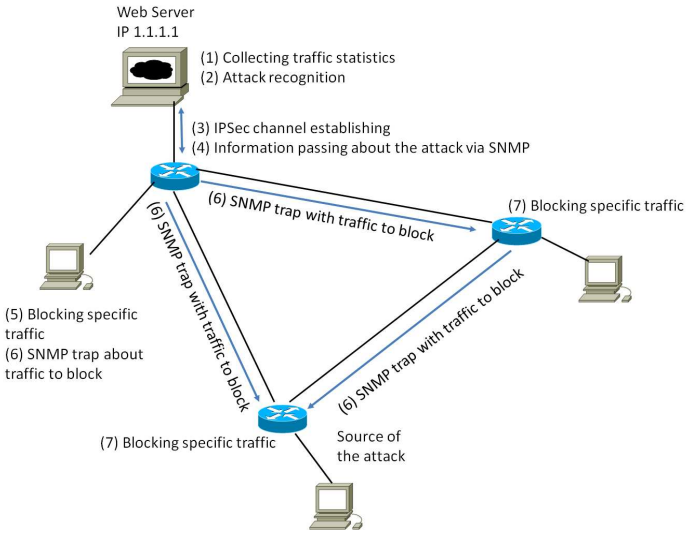

Figure 3: The algorithm used

The algorithm for dropping packets on routers and firewalls are using well known QoS method idea with some additional modules. In QoS methods network bandwidth is managed using special algorithms. This algorithms allow to set a given bitrate in the network for specific users or type of network traffic. In this manner a privileged traffic is selected. For the traffic not classified as privileged, usually the Fair Queue method is applied. This method allows fair bandwidth division between data sources. Nowadays it is widely used on routers and firewalls. It should be noted that when bandwidth is limited in the receiver's network, this method divides packets fairly so that every stream can reach the receiver. In the method a weighted traffic mechanism is also introduced. The difference is that it is not basing on data source or category, but on connection history, especially server network statistics. What should be noticed is that during normal network activity, data streams are transmitted to the receiver over a router or firewall. When an attack occurs, the number of streams grows rapidly up to the maximum number supported by the network resources.

When the number of packets exceeds the given limits, the proposed method has to remove them from the queue by using a well-known mechanism called Random Early Detection (RED). This RED mechanism is widely implemented on routers and firewall also. In a standard method, the RED mechanism is executed on incoming packets with data streams categorization. In the proposed method, the RED mechanism should not be executed on the data streams that had been transmitted earlier. That is why the method has to:

- keep the data streams history from the moment before the attack and this period should not be neither too long, as otherwise it will not capture the

appropriate traffic, nor too short, in order to protect the device from heavy load,

- after detecting the attack, set the traffic filtering rules to privilege historical packets, i.e. by selecting them with the use of filtering rules on a router module,
- block all other malicious traffic, i.e. by using RED mechanism [Changwang, 10] or firewall rules [Cheswick, 94][Chapman, 95] but in intelligent way.

Figure 4 shows a general method scheme.

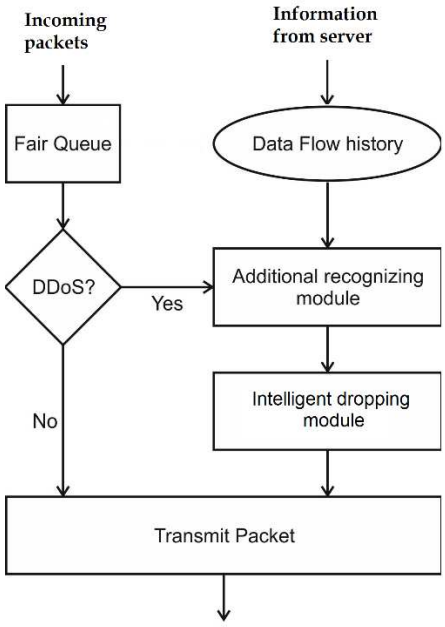

Figure 4: The algorithm used

2.3 The fuzzy adaptive dropping algorithm

An intelligent dropping module was implemented as fuzzy adaptive dropping algorithm. In the original dropping module a special firewall dropping module was developed, the role of which is to filter the traffic on special server's open port and to limit it according to the determined policy in intelligent way. During the server's regular work all packets are passed through the module. The regular work means that the network is stable and not under any attack. When the attack is recognized and the filtering module is informed about it, the intelligent dropping module start his work.

When the process started the list of the IP addresses for the filtration is collected from the server. This set of IP addresses consist these which were operating with the server in right way and the system knows that they are not a part of the packets used for the attack – *correctIPlist.* The filtration process has to validate if the IP source of the packet is on the *correctIPlist.* If the packet is on these list, means that it could be

passed through without any delays. In other case, the algorithm is checking if the number of passed packed *passed_counter* is lower that the limit provided by the administrator *limit_counter* in a current time slot *t1*. If the *limit_counter* is exceeded in the current time slots *t1*, the algorithm has to drop this packet. In the next time slot the *passed_counter* is set to 0, so the process could start to filtering packet again. The module responsible for dropping packets has to regulate the opening connection limit in intelligent way, which are as follows:

- the *limit_counter* could be decreased when during some subsequent time slots *time_slots* the limit of the packets was exceeded, this situation gives an information, that server is under huge attack and the system should give him a chance to release some resources;
- the *limit_counter* could be increased when during some subsequent time slots *time_slots* the limit of the packets was not exceeded, this situation gives an information, that server is not under huge.

The problem of such a solution was the fact that it is insensitive to the current trend. In the case of an ongoing attack, the packet limit should not be increased too quickly, whereas in the case of an expiring attack, this limit can be increased faster, allowing users to gain access to services.

For this purpose, an fuzzy adaptive algorithm was developed using Oriented Fuzzy Numbers. The proposed algorithm measures the number of connections for subsequent periods of time slots:

$$t_i, t_{i-1}, t_{i-2}, t_{i-3} \tag{1}$$

Where t_i is a current timeslot.

All four measures together give an fuzzy number in OFN notation presented in figure 5, where:

- $f_A(0)$ responds to t_{i-3},
- $f_A(1)$ responds to t_{i-2},
- $g_A(1)$ responds to t_{i-1},
- $g_A(0)$ responds to t_i.

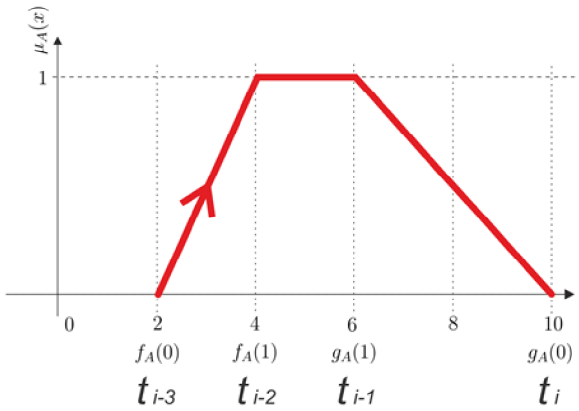

Figure 5: Fuzzy number in OFN notation.

Def. 1
Fuzzy observance of Firewall device in time t_i is a set

$$Firewall/t_i=\{f_A(0)/t_{i-3}, f_A(1)/t_{i-2}, g_A(1)/t_{i-1}, g_A(0)/t_i\} \quad (2)$$

where

$$t_i > t_{i-1} > t_{i-2} > t_{i-3} \quad (3)$$

$$|t_i - t_{i-1}| = |t_{i-1} - t_{i-2}| = |t_{i-2} - t_{i-3}| = t_n \quad (4)$$

timeslot of the measurement

$$f_A(0) \le f_A(1) \le g_A(1) \le g_A(0) \quad (5)$$

Lemma 1

$$Firewall_{positive}=f_A(0) < f_A(1) < g_A(1) \text{ or } f_A(1) < g_A(1) < g_A(0) \quad (6)$$

Otherwise $Firewall_{negative}$.

According to this definition, what we obtain is:

- OFN with positive order when the packet count increases,
- OFN with negative order when the packet count decreases.

The analysis of the packet statistics, along with the appropriate counters giving a fuzzy number.

When we get $Firewall_{positive}$ we know, that the number of connections are still growing. When we get $Firewall_{negative}$ we know, that the number of connections are decreasing.

So the algorithm was modified an working as follows: when the process started the list of the IP addresses for the filtration is collected from the server. This set of IP addresses consist these which were operating with the server in right way and the system knows that they are not a part of the packets used for the attack – *correctIPlist.* The filtration process has to validate if the IP source of the packet is on the *correctIPlist.* If the packet is on these list, means that it could be passed through without any delays. In other case, the algorithm is checking if the number of passed packed *passed_counter* is lower that the limit provided by the administrator *limit_counter* in a current time slot *t1*. If the *limit_counter* is exceeded in the current time slots *t1*, the algorithm has to drop this packet. In the next time slot the *passed_counter* is set to 0, so the process could start to filtering packet again. The module responsible for dropping packets has to regulate the opening connection limit in intelligent way, which are as follows:

- during time slots the OFN number Firewall is calculated,
- when the $Firewall_{positive}$ the limit of packets *packet_limit allowed* to passed in time slots is decreased,
- when the $Firewall_{negative}$ the limit of packets *packet_limit allowed* to passed in time slots is increased.

This is a very important case, because it allows the server to handle the incoming connections which may be potentially correct or to release the resources used incorrectly by the attacker. Despite the attack, the server is still accessible to the users who were working on it when the attack was detected.

3 The method test results

3.1 Test environment

The presented method was implemented and tested the same way as in [Apiecionek, 16]. The same network were build and the same test procedure were made using proposed fuzzy adaptive dropping algorithm. This module was implemented for the firewall Iptables in the Debian Linux system based on kernel 2.6.32. The test network is presented in figure 6. During the test there were a simulation of common attack on server. As a server was used Asterix FreePBX distribution on CentOS with kernel 2.6.32 with the web server Apache 2.2.15. This server was equipped in 1GB RAM. As a hacker system was used DDOSIM software (Layer 7 DDoS Simulator v 0.2) also on Debian Linux with kernel 2.6.32 with 512 MB RAM. The connection between server and network was protected with the firewall with proposed method implemented with fuzzy adaptive dropping algorithm. This firewall was on Linux Debian with kernel 2.6.32 on machine with 512 MB RAM.

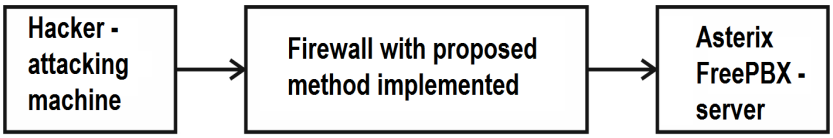

Figure 6: The test network [Apiecionek, 16].

The observation of the attack consist of memory usage and web server response time. The attack construction consist of sending the huge HTTP/get message to the web server which has to allocate resources for answering. When the proposed method was not used, the server's memory usage was on 100% level and the server stopped answering on user requests. When the proposed method was implemented such situation does not occur. The tests has five different conditions, but all of them have lasted one hour. These five conditions have different amount of HTTP/get requests sent with different delays between them. These amounts and delays were as follows:

- 100 amount of HTTP/get requests with the delay 30 seconds;
- 1000 amount of HTTP/get requests with the delay 30 seconds;
- 2000 amount of HTTP/get requests with the delay 10 seconds;
- 10000 amount of HTTP/get requests with the delay 10 seconds;
- 50000 amount of HTTP/get requests with the delay 30 seconds;

3.2 Test results

The test results were compared with the result achieved with previous dropping module [Apiecionek, 16]. RAM usage on the http server and on the firewall with the old method during the test is presented in figures 7, while with the fuzzy adaptive one in figure 8.

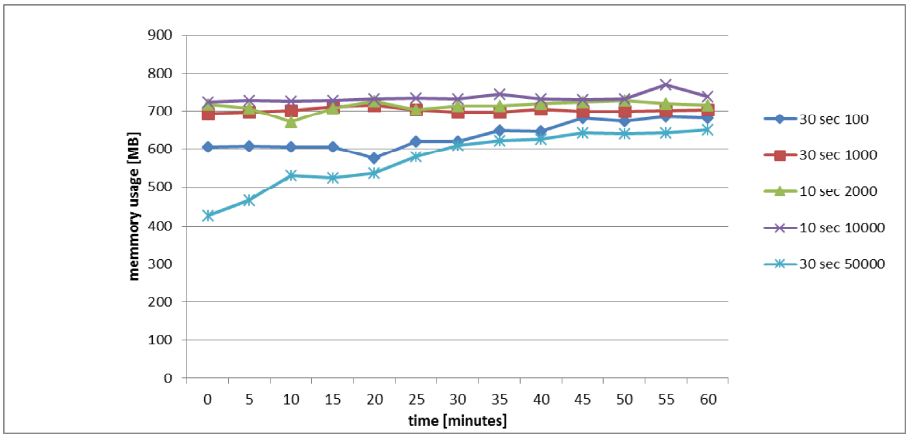

Figure 7: RAM usage on http server with the old method [Apiecionek, 16]

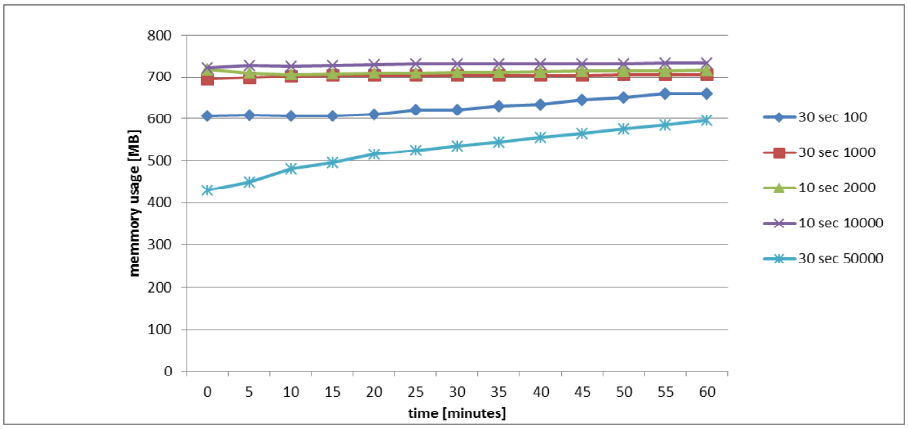

Figure 8: RAM usage on http server with the fuzzy adaptive method.

Figures 9-13 presents the direct comparison of the five tests made.

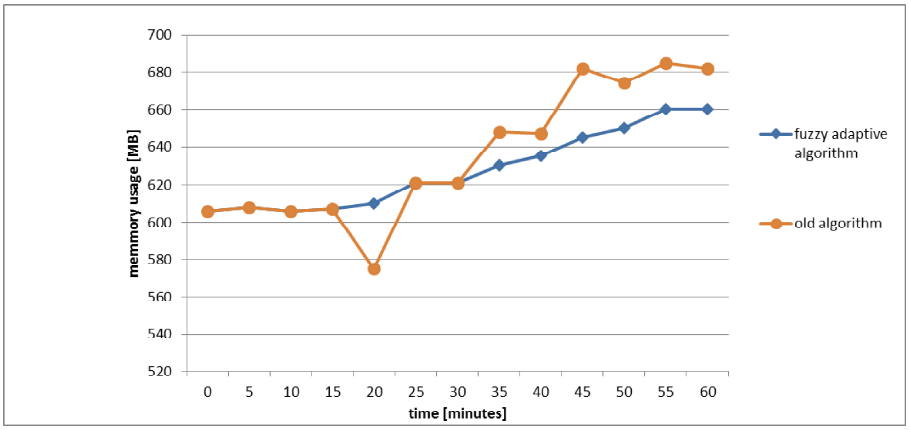

Figure 9: RAM usage on http server during the test 100 HTTP GET requests sent every 30 seconds.

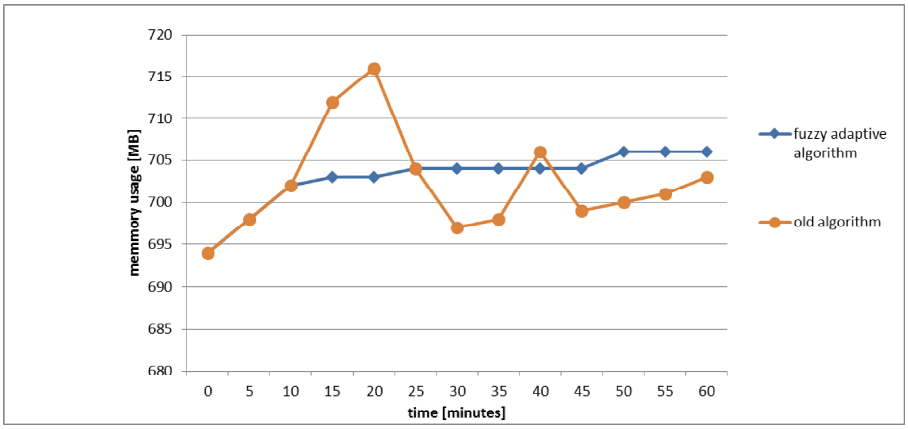

Figure 10: RAM usage on http server during the test 1000 HTTP GET requests sent every 30 seconds.

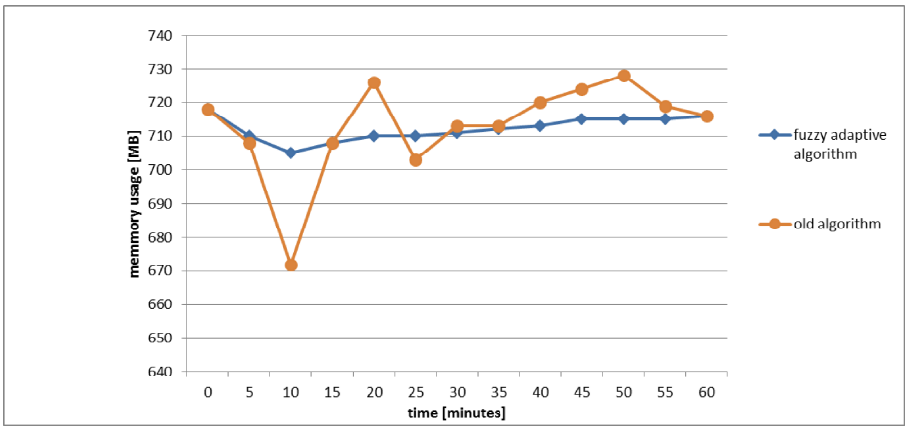

Figure 11: RAM usage on http server during the test 2000 HTTP GET requests sent every 10 seconds.

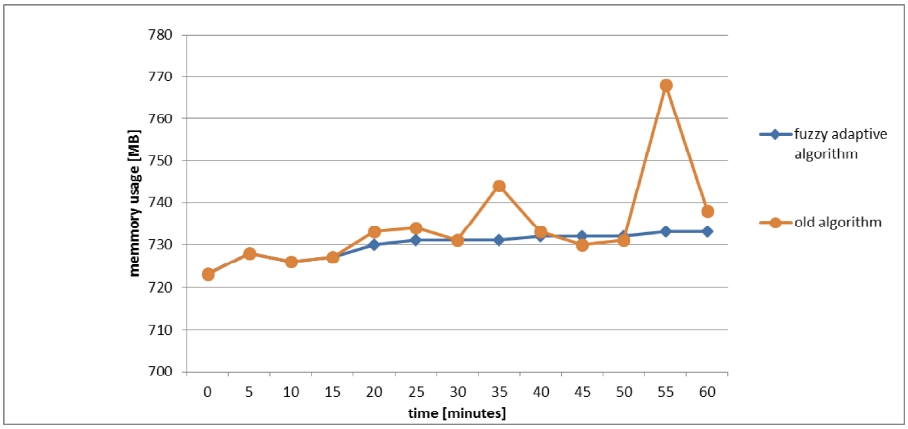

Figure 12: RAM usage on http server during the test 10000 HTTP GET requests sent every 10 seconds.

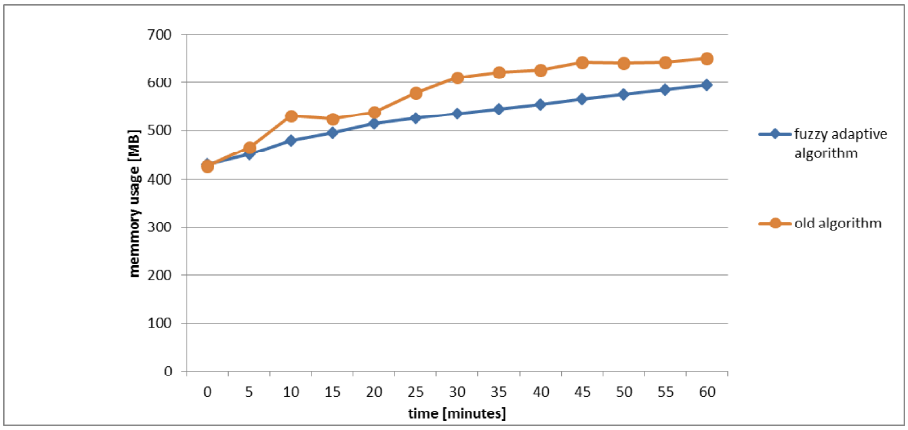

Figure 13: RAM usage on http server during the test 50000 HTTP GET requests sent every 10 seconds.

As it could be noticed, the first 4 measurements give almost the same results. Then when the Ordered Fuzzy Number is calculated, the algorithm begin create the stable condition for the server work. The memory usage is more stable than in the test with the previous algorithm used. So, the application working on the server are also more save, because the memory allocation could be predicted in better way.

4 Conclusions

The new fuzzy adaptive algorithm for fighting with DDoS attack for IoT systems was presented in this article. Using the Ordered Fuzzy Numbers give possibility to predict the situation according to the trends. It lets to predict if the attack will grow up or will be finished. In the method presented in the article the algorithm lets to set server memory usage on more stable level. The same results could be achieved with Ordered Fuzzy Numbers also in different needs for example:

- better could servers management according to their usage [Apiecionek, 17A],
- limiting energy consumption by decreasing packets retransmissions in 5G network [Apiecionek, 17B],
- faster DDoS attack recognition [Apiecionek, 17C].

What is worth to mention is that this fuzzy adaptive algorithm do not change any positive feature of the whole method used in the test proposed by Apiecionek et. al [Apiecionek, 16].

The comparison of the proposed method with the existing others is presented in Table 1. This comparison is made according to server memory usage, possibility of initiating connection during attack, connection limit after the attack stops, required administrator task to do after the attack.

In the future work the algorithm should be informed by the server about his characteristics and the method should use optimal way of limiting packets [Kozik, 16] [Andrysiak, 16].

The idea of proposed method could be used for Critical Infrastructure Protection and IoT systems protection. The place for proposed method installation is shown in figure 14 by the red star. This method should be considered to become some RFC standard in future. Without such standards the future could be tragic. The presented problem should be considered widely nowadays. If not, the DDoS attacks will become much bigger problem in the future network than they are today.

The author declares that there is no conflict of interest regarding the publication of this paper.

Criteria of the comparison	**Literature methods**	**Old method**	**Fuzzy adaptive method**
memory usage of the server	almost 100% during the attack	Constant on defined amount during the attack	Almost stable value
Possibility for using server during attack	No response to the normal user	Response to the normal user who were working before the attack	Response to the normal user who were working before the attack
Possibility for the new connection to the server when the attack occurs	Impossible during the attack when the whole packets are dropped by the firewall.	Possible, Network browsers tries to connect multiple time, so the user get the chance to connect to the server.	Possible, Network browsers tries to connect multiple time, so the user get the chance to connect to the server.
Possibility for user to finish work during attack	Impossible – the firewall are blocking all traffic.	Possible by the user who were connected to the server before the attack started.	Possible by the user who were connected to the server before the attack started.
The number of connections during attack	Zero – the firewall are blocking all traffic.	Limited to defined amount by administrator according to proposed algorithm.	Limited to defined amount by administrator according to proposed algorithm.
The number of connections after the attack	Need administrator work to get back.	No work of the administrator is required to back to previous state.	No work of the administrator is required to back to previous state.
Required administrator for network reconfiguration after the attack	Required	Not required	Not required

Table 1: Method comparison

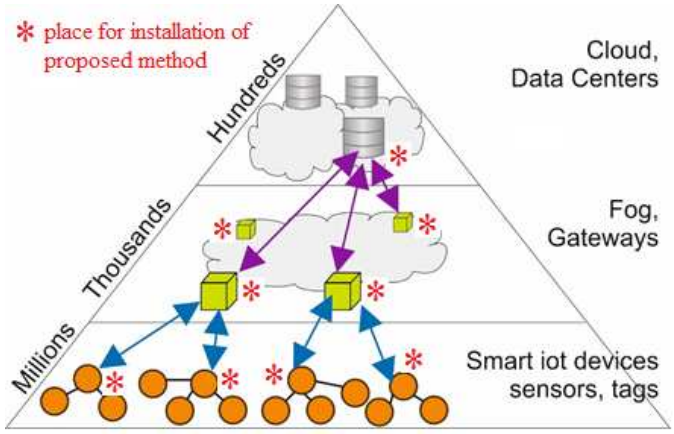

Figure 14: A place for installation proposed method in the IoT systems.

References

[Al-Fuqaha, 15] Al-Fuqaha, A.I., Aledhari, M., Ayyash, M., Guizani, M., Mohammadi, M.: (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys and Tutorials, 17, 2347-2376

[Ambak, 09] K. Ambak, R. Atiq, R. Ismail: Intelligent Transport System for Motorcycle Safety and Is-sues, European Journal of Scientific Research, ISSN 1450-216X Vol.28 No.4 (2009), pp. 600-611

[Andrysiak, 16] Andrysiak T., Saganowski L., Choras M.: Kozik R.: Proposal and comparison of network anomaly detection based on long-memory statistical models , Logic Journal of the IGPL, vol. 24, issue 6, 944-956, 2016.

[Andrysiak, 2020] Andrysiak T., Saganowski Ł.: Anomaly Detection for Smart Lighting Infrastructure with the Use of Time Series Analysis, Journal of Universal Computer Science, In: Vol. / Issue 4, 2020

[Apiecionek, 14] Apiecionek Ł., Czerniak J. M., Zarzycki H.: Protection Tool for Distributed Denial of Ser-vices Attack, "Beyond Databases, Architectures, and Structures", 405-414, 2014, Springer International Publishing

[Apiecionek, 16] Apiecionek Ł., Makowski W.: Intelligent FTBint Method for Server Resources Protection, Beyond Databases, Architectures and Structures. Advanced Technologies for Data Mining and Knowledge Discovery, Volume 613 of the series Communications in Computer and In-formation Science pp 683-691, 2016.

[Apiecionek, 17A] Apiecionek Ł., Czerniak J., Dobrosielski W., Ewald D.: Fuzzy Logic Load Balancing for Cloud Architecture Network - A Simulation Test, Advances in Fuzzy Logic and Technology 2017. Proceedings of: EUSFLAT-2017 - The 10th Conference of the European Society for Fuzzy Logic and Technology, September 11-15, 2017, Warsaw, Poland IWIFSGN'2017 - The Sixteenth International Workshop on Intuitionistic Fuzzy Sets and

Generalized Nets, September 13-15, 2017, Warsaw, Poland, Volume 1 / ed. Janusz Kacprzyk, Eulalia Szmidt, Sławomir Zadrożny, Krassimir T. Atanassov, Maciej Krawczak., Cham: Springer International Publishing, 2017, pp. 43-54

[Apiecionek, 17B] Apiecionek Ł.: Limiting Energy Consumption by Decreasing Packets Retransmissions in 5G Network, Mobile Information Systems, vol. 2017, Article ID 4291091, 9 pages, 2017. doi:10.1155/2017/4291091

[Apiecionek, 17C] Apiecionek Ł.: Fuzzy Observation of DDoS Attack, Theory and Applications of Ordered Fuzzy Numbers. A Tribute to Professor Witold Kosiński / ed. Piotr Prokopowicz, Jacek Czerniak, Dariusz Mikołajewski, Łukasz Apiecionek, Dominik Ślęzak. Cham: Springer International Publishing, 2017, Studies in Fuzziness and Soft Computing, 1434-9922; 356, pp. 241-254

[Bernieri, 17] Bernieri G., Etcheves Miciolino E., Pascucci F., Setola R.: Monitoring system reaction in cyber-physical testbed under cyber-attacks, Computers & Electrical Engineering 59 (2017): 86-98

[Bujnowski, 2020A] Bujnowski S., Marciniak T., Marciniak B., Lutowski Z., The Analysis of the Possibility to Construct Optimal Third-degree Reference Graphs, Journal of Universal Computer Science, In: Vol. 26 / Issue 4, 2020

[Bujnowski, 2020B] Bujnowski S., Marciniak T., Marciniak B., Lutowski Z.: Impact of Resource Control in Irregular Networks on their Transmission Properties, Journal of Universal Computer Science, In: Vol. 25 / Issue 6, 2020

[CERT, 00] CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks, November 2000, http://www.cert.org/advisories/CA-1996-21.html

[CERT, 97] CERT Advisory CA-1996-01 UDP Port Denial-of-Service Attack, September 1997, http://www.cert.org/advisories/CA-1996-01.html

[Chapman, 95] Chapman B., Zwicky E.D.: Building Internet Firewalls, O'Reilly & Associates, Inc., 1995, ISBN 1-56592-124-0

[Changwang, 10] Changwang Z., Jianping Y., Zhiping C. Weifeng Ch.: RRED: Robust RED Algorithm to Counter Low-Rate Denial-of-Service Attacks, IEEE COMMUNICATIONS LETTER S, VOL. 14, NO. 5, MAY 2010.

[Cheswick, 94] Cheswick W.R., Bellovin S. M.: Firewalls and Internet Security: Repelling the Wily Hacker, Addison-Wesley Publishing Company, 1994, ISBN 0-201-63357

[Choras, 16] Choras M., Kozik R., Flizikowski A., Renk R., Holubowicz W.: Cyber Threats Impacting Critical Infrastructures , in: Setola R. et al.: Managing the Complexity of Critical Infrastruc-tures, Studies in Systems, Decision and Control, vol. 90, 139-161, Springer, 2016.

[D'Antonio, 06] D'Antonio, S., Oliviero, F., & Setola, R.: (2006, August). High-speed intrusion detection in support of critical infrastructure protection. In International Workshop on Critical Infor-mation Infrastructures Security (pp. 222-234). Springer, Berlin, Heidelberg

[Knowles, 15] Knowles W., Prince D., Hutchison D., Jules Pagna Disso J.F., Jones K.: A survey of cyber security management in industrial control systems, International journal of critical infrastructure protection 9 (2015): 52-80

[Kozik, 16] Kozik R., Choras M., Holubowicz W.: Evolutionary-based packets classification for anoma-ly detection in web layer , Security and Communication Networks, vol. 9, Issue 15, 2901-2910, Wiley, 2016

[Kubilay, 18] Kubilay D., Hatem I., Vateva-Gurova T., Neeraj S.: Securing the Cloud-Assisted Smart Grid, International Journal of Critical Infrastructure Protection (2018), doi: https://doi.org/10.1016/j.ijcip.2018.08.004

[Mazur, 2019] Mazur D., Paszkiewicz A., Bolanowski M., Budzik G., Oleksy M.: Analysis of possible SDN use in the rapid prototyping process as part of the Industry 4.0, BULLETIN OF THE POLISH ACADEMY OF SCIENCES TECHNICAL SCIENCES, Vol. 67, No. 1, 2019, DOI: 10.24425/bpas.2019.127334

[Moor, 06] Moor D., Shannon C., Brown D.J., Voelker G. M.: Savage S. Inferring Internet Denial-of-Service Activity, ACM Transactions on Computer Systems (TOCS) 24 (2), 115-139, 2006.

[Osanaiye, 16] Osanaiye, O., Choo, K. K. R., & Dlodlo, M.: (2016). Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework. Journal of Network and Computer Applications, 67, 147-165

[Piechowiak, 12A] Piechowiak M., Zwierzykowski P.: "The Evaluation of Unconstrained Multicast Routing Algorithms in Ad-hoc Networks", The International Science Conference: Computer Net-works CN2012, Szczyrk, Poland, 2012.

[Piechowiak, 12B] Piechowiak M., Zwierzykowski P.: "The Evaluation of Multicast Routing Algorithms with Delay Constraints in Mesh Networks", 8th IEEE, IET Int. Symposium on Communication Systems, Networks and Digital Signal Processing CSNSDP 2012, Poznań, Poland, 2012

[Pietkiewicz, 2018] Pietkiewicz P., Nalepa K., Miąskowski W., and Wilamowska-Korsak M.: A system for monitoring and controlling a thermal energy store and an energy capture system, BULLETIN OF THE POLISH ACADEMY OF SCIENCES TECHNICAL SCIENCES, Vol. 66, No. 6, 2018 DOI: 10.24425/bpas.2018.125942

[Renk, 11] Renk R., Choras M., Saganowski Ł., Holubowicz W.: Signal Processing Methodology for Network Anomaly Detection, in: Intrusion Detection Systems, Pawel Skrobanek (Ed.), ISBN: 978-953-307-167-1, InTech, 2011.

[Rocky, 02] Rocky K., Chang C.: Defending against Flooding-Based Distributed Denial-of-Service At-tacks: A Tutorial, IEEE Communications Magazine, October 2002, pp. 42-51

[Schuba, 96] Schuba C. L., Krsul, I., Huhn M. G., Spafford E. H., Sundaram A.: Analysis of a Denial of Service Attack on TCP (1996). Computer Science Technical Reports. Paper 1327. http://docs.lib.purdue.edu/cstech/1327

[Sundmaeker, 10] Sundmaeker, H. P. F., Guillemin P., and Woelfflé, S.: Vision and Challenges for Realising the Internet of Things. Pub. Office EU, 2010 [Online]. Available: http://www.internet-of-thingsresearch.eu/pdf/IoT_Clusterbook_March_2010.pdf

[Ten, 10] Ten, C. W., Manimaran, G., & Liu, C. C.: (2010). Cybersecurity for critical infrastructures: Attack and defense modeling. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 40(4), 853-865