ISSN 2347 - 3983

Volume 8. No. 9, September 2020

International Journal of Emerging Trends in Engineering Research

Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter220892020.pdf

https://doi.org/10.30534/ijeter/2020/220892020

Duplicate Packet Detection in the Multicast Network using Efficient Packet sequencing approach at Mobile Receiver

Jagadeesha R[1] Thippeswamy K[2]

[1]Department of Computer Science & Engineering, Kalpataru Institute of Technology, Tiptur, VTU, Belagavi, Karnataka, India. jagdish.mtech@gmail.com

[2]Department of Computer Science & Engineering,VTU PG Center Mysore, Karnataka, India. thippeswamy_k@hotmail.com

ABSTRACT

Duplicate packet detection during signature verificatiapproach, suitable for static receivers. In our proposed packet sequence number approach, packet sequence number is calculated by random number and constant value at the multicast sender. This constant value is shared with all the receivers before the start of transmission. The size of the reserved bytes for storing packet sequence number is modified or unchanged based on the available energy at the receiver. The energy is calculated by a multicast receiver and forwarded to a sender during signature verification. The sender updates reserved bytes for storing sequence number accordingly, which reduces unnecessary usage of memory and processing of sequence number.

Key Words: duplicate packet detection, energy, mobile receiver, packet sequence number.

1. INTRODUCTION

In the multicast network [1] and information system, data duplication is a generic problem. In network analysis data duplication means duplicate packet [2]. With the increased use of multicasting and MANET for different purposes for a range of devices such as mobile, computer, etc. multiple gateways are used within multicast or mobile network posses some advantage over a single gateway in a network in terms of packet loss. If the packet is lost in the network due to congestion or any other reason, other gateways in the network, which has the copy of the packet will try to deliver the packet successfully. Usage of multiple gateways in the network itself creates an unnecessary load on the node to check the duplicate packets which are generated at different gateways. Switches, routers, etc., implement the duplicate packet monitoring in high speed links. Network devices can maintain the information about packets that they have received and forwarded for a period of time. When network devices detect the duplicate at any node or device makes use of resources: bandwidth, memory and CPU, these resources are rich in multicast static receivers but poor in multicast mobile receiver [3] [4]. Duplicate packets are on the devices are identified by information available in the packet header, but the duplicate packet generated in the digital signature header and at the payload is not identified by the packet header because in the digital signature separate signature header is attached to the packet header which will not be processed or detected by the devices because the same payload and signature present in two different packets. So separate scheme or method is needed to identify the duplicate packets at the signature header is required. (DPD) Duplicate packet detection is very much necessary in MANET [5], wireless and multicast network. Packets may be transmitted out via the same interface as the one where they received or from the different interface. Routers and switches may also receive the same copy of the packets from different neighbors. A temporal packet identification mechanism is recommended to detect and reduce the duplicate packet forwarding where incoming packets are compared with packets which are present in the buffer and recently processed. The packets can be duplicated at different stage switches, routers, Network address translation and at transport process are identified and processed by the device itself. Different schemes and methods are available for duplicate packet identification and removal or avoiding but separate scheme for energy constraint duplicate packet detection scheme is needed for Mobile receivers. Scheme in [6] utilizes energy efficiently, suitable for mobile receiver. In this paper, we introduce the packet sequence number of different sizes based on the available energy and the batch size in the signature header of the batches. Our research paper is organized as follows: part 1: focused on the introduction and related works and drawbacks. Part 2: proposes the basic idea and research method. Part 3: compares the results and findings of research carried out. Part 4: highlights the conclusion of the proposed work.

1.1 Related Work

In some situations a forged packet from the attacker or from intermediate devices disrupts batch signature verification, which leads to DoS (Deniel of Service) [7] attack. In [1] mark is applied to each packet recursively using divide and conquer approach, which is unique to every packet and not easily spoofed. This method is used to identify the duplicate packet in static receivers due to its high computation and processing cost not suitable for mobile receivers. In [8] [9] [10] tolerate DoS attack to maximum level, here duplicate packet injection [11] [12] is controlled by using constant factor. Computation cost is less when compared to [1], but processing of constant factor to each packet creates extra overhead to the receiver not suitable for mobile receivers. In [13] numbers of gateways in the network are minimized to some level to reduce the maximum number of duplicate packets. In [13] uses a unique packet identifier at the gateway using a hash function to identify the packet which is forwarded from each gateway, minimizes complexity and the traffic due to duplicate packet occurrence in a multicast network. The main drawback of [13] is the size or number of bytes reserved to store the packet unique identification is calculated using a hash and effective utilization of energy or resource of mobile receiver is not addressed. Duplicate packet detection methodology in [14] uses sliding window [15] method and packet comparison process. In sliding window method, each packet is compared with all other packets in the window size n, if the packet matches then it is marked as duplicate. Adjusting the size of the window when the load on the receiver is varied is not discussed. Packet comparison in [15] each packet is compared with tcp/udp payload, IP payload and Ethernet payload are compared first, if any of the fields matches than type of duplicate is identified. These identifications can be done by the device itself, so no separate scheme is required, which creates an unnecessary burden to mobile receiver resources. In [16] uses two methods for duplicate packet detection: sequence number duplicates packet detection (S-DPD) and Hash based duplicate packet detection (H-DPD). S-DPD minimizes memory and CPU required to duplicate packet identification, H-DPD minimizes bandwidth required to duplicate packet identification. We introduce packet sequence number identification technique for energy constraint mobile devices [17] [18] [19] during batch signature verification. [20] Uses MD5 and SHA-1 methods distribute data across a variety of cloud servers for duplication which increases reliability to some extent. The selective data compression method is used for removing same copy of data. This duplication is not suitable for duplicates packet detection during signature verification because signature verification and duplicate packet identification are carried out on a same node. We are considering multicast mobile node which has the low processing capability, so it's not possible to share the packets across the different servers or nodes.

2. RESEARCH METHOD

Here we are using efficient packet sequence number approach for packet identification during signature verification.

2.1 Basic Idea

When a multicast packet is transferred from a sender to group of receiver in the network [21] [22] [23], some nodes are static having enormous energy and others having limited capability mobile node. Different types of digital signature schemes are available for multicast batch signature authentication. MABS [1] provides a mechanism to overcome from DoS attack and duplicate packets. Duplicate packet detection in [16] S-DPD identifies duplicate packet by inserting a packet sequence number to each packet. Packet identification, length used is fixed in the header of the packet, but the reserved bytes for storing sequence of the packet are not increased or decreased based on the number of packets in different batches to be considered. Some application like video streaming and online stock exchange, etc., uses more than thousand packets processing at a particular time to play out. Two bytes reserved for storing packet sequence number is not sufficient to store identity. Two or more packets may get the same sequence in some instance for that either we have to reset the packet sequence number of the sender memory to zero after some traffic or the number of bytes reserved for storing the sequence number in the header of the packet to be increased. If we increase the size of the reserved bits in the header for storing sequence number, creates wastage of memory bytes when not in use and processing of large bits in the packet header leads to unnecessary utilization of energy at the receiving node. If we reset the reserved bytes used for the sequence number to zero in the packet header than on what factor it will be done. If it resets automatically than how the receiver will come to know when the sequence number is set to zero, even if we calculate also it needs extra monitoring along with the comparison of sequence number both at the sender and receiver. So algorithm or method is required for sequence number insertion in the packet header [24] and setting sequence number in memory to zero based on the current situation. Increasing or decreasing reserved bytes for storing the sequence number in the header of the packet at the sender upon request from a receiver. This processing requires some extra energy and resources that can be managed in our proposed methodology.

2.2 Proposed Methodology

In our proposed method packet identification is calculated and inserted at the sender using constant factor and sequence number which is shared with the receiver. Receiver also calculates the packet identification using the constant factor and sequence number received from the sender and compares with the each packet and store in memory

2.2.1 Sequence number insertion at the sender

Sender forwards random constant K and the first sequence number Si to the receiver, first three bits of two bytes are used to store this constant value. Sequence number **S** starts from the lowest possible value usually less than ten and stored in remaining fifteen bits of two bytes. Constant factor 'K' and first sequence number 'S1' is calculated by using random function, which is then converted to numbers within 20 by using modulo operation.

K_{randm}= random-function() (1)

Equation (1) calculates random number stores to K_{randm}

K= K_{randm} % 20 (2)

Equation (2) updates the value of K_{randm} to less than 20 and stores in K

$S1_{randm}$= random-function() (3)

Equation (3) calculates random number stores to $S1_{randm}$

S1= $S1_{randm}$ % 20 (4)

Equation (4) updates the value of $S1_{randm}$ to less than 20 and stores in S1

$0 < K < 19$ and $0 < S1 < 19$

K and S1 value should be greater than zero and less than 19, if either value is equals to zero than discards it and recalculates S1 and K.

Sequence number identification is inserted into the header of each packet at the time of signature generation.

$$\int_{i=1}^{n} Sseq = \int_{i=1}^{n} Si + k \qquad (5)$$

Equation (5) calculates Sseq for each packet from 1 to n, each Sseq is sum of Si and K. Sseq is the sequence number is calculated for packet identification to each packet in the batches. *'n'* is the number of packets in the batch and calculates sequence number Si+K to each packet in the batches using the above equation and added in the packet header. 'K' and Si value are recomputed and shared with the receiver when the reserved bytes for storing sequence number modified. Signature is applied to each batch. Apply the above method to all the packets in the batches.

When the receiver requested to increase the batch size than the sender check the reserved byte for packet id, if its two bytes than a reserved byte of packet id is double that is four, if it's four bytes than sender keeps four byte itself as the reserved byte for packet id. When the receiver requested to decrease the batch size than the sender checks the reserved byte for packet id, if it's four bytes than a reserved byte of packet id is reduced to half the original that is two. If its two bytes than sender keeps two byte itself as the reserved byte for packet id.

Consider the batch 'B' contains 'n' number of packets at the Sender 'S'

Applying sequence number during Signature generation

sequence-number (signed(B)) (6)

Equation (6) applies sequence numbers of all packets to the signature header during signature generation than transferred to multicast receiver R.

2.2.2 Sequence number verification at the receiver

Receiver checks available energy at the receiver for every 't' second time interval to update the sender to modify the reserved bytes for storing the sequence number in the signature header. When the available energy is more than 50% than receiver will request one time to double the reserved bytes for storing sequence number. When the available energy is less than 50% than receiver will request one time to reduce the reserved bytes for storing sequence number to half the original value. When a signed batch reaches the receiver, the receiver verifies the signature.

E_{max} → maximum energy of the node

E_t → energy at time 't'

E_t = Energy-calculate(R) (7)

Equation (7) calculates the energy available at the receiver and stores in E_t

$E_t > E_{max}/2$ (8)

Equation (8) checks whether the value of E_t is greater than half of the maximum available energy at the receiver than receiver request to double the reserved bytes for the sequence number

$E_t <= E_{max}/2$ (9)

Equation (9) checks whether the value of E_t is less than half of the maximum available energy at the receiver than receiver request to reduce the reserved bytes for the sequence number to half the original value

The receiver calculates Si+K. of the each packet, and stores in the memory. For every Si+K presents in the two byte header of receiving packets and signature header present in each batch is compared with all the Si+K present in the memory, if there exists any duplicate Si+K than that packet id is considered as duplicate and discards it. Si+K in the memory is removed when a new 'S' and 'K' is shared from the sender, this reduces the number of comparisons carried for Si+K between memory and signature header.

Sequence number identification is calculated by the receiver for comparing the sequence number of the signature header and stores in the memory.

$$R_{seq} = \int_{i=1}^{n} Si + k \qquad (10)$$

Equation (10) calculates the sequence number from 1 to n

Signature verification function

Batch-signature-verify (Signed(B)) (11)

Equation (11) verifies signature, receiver calculated sequence number 'R_{seq}' is compared with the 'S_{seq}' in signature header of the batch, if there is any duplicate sequence number has come than the packet with that sequence number is discarded. Sequence number verification is passed during signature verification than data presented in the batches are displayed without duplication.

3. RESULT AND DISCUSSION

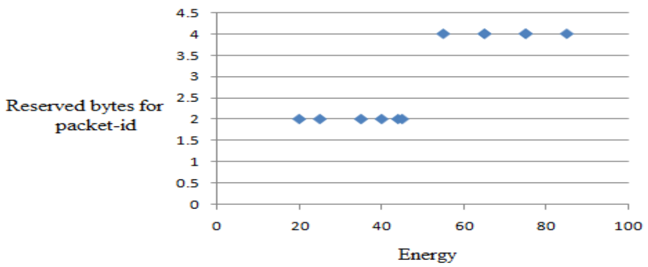

Figure 1: Reserved bytes for packet-id based on the energy

Figure 1. Shows reserved byte for storing packet identification using the sequence number method. When the energy of the receiver is reached to less than 50 percent than the reserved byte for storing packet identification is reduced to two and its updated to four when the energy at the receiver crosses to more than 50 percent. This updating the reserved bytes for storing packet identification creates negligible overhead to sender and receiver, but significantly helps for mobile receiver which has low energy and capability at all the time.

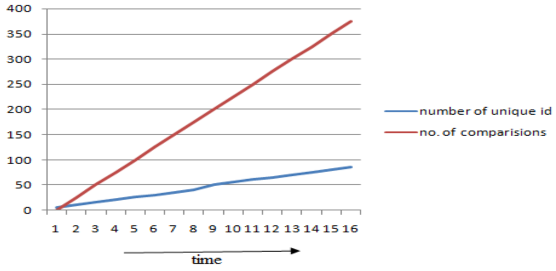

Figure 2: More number of packet identity comparison

Figure 2. Shows number of comparisons to check for duplicate packet identification increases at the receiver for each new packet of the batches in all existing methods. At times t=1 to t=14 number of comparisons for packet identification is increased to 375 from 1 and unique packet identification is also increased to 85 from 1. If the packet identification not set to zero after some interval than the number of comparisons staring at the receiver is increased significantly. This comparison increases to thousands together in live streaming video which are sent through batches. This comparison time becomes more than the signature verification time and utilizes maximum energy of the receiver not suitable for energy constrained mobile receivers.

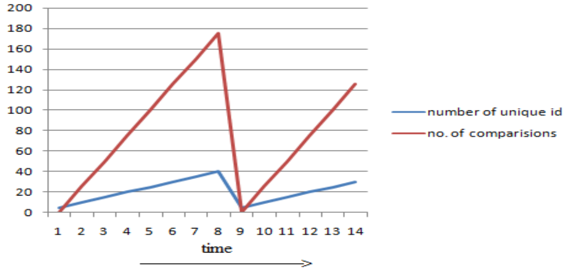

Figure 3: Less number of packet identity comparison

Figure 3. Shows the number of comparisons to check for duplicate packet identification increases at the receiver for each new packet of the batches. At time t=1 to t=8 number of comparisons for packet identification is increased to 180 from 1 and unique packet identification is also increased to 40 from 1. When the time t=9 receiver requested to update the reserved bits in the header for packet identification (either two or four) than the sender starts calculating unique packet identification from the 1 updating its memory to 0. The receiver also updates its memory for storing packet identification after comparison to zero and starts a new entry of unique packet identification in the memory. Hence, in the figure at time t=10 packet identification comparison again starts from 1. If the packet packet identification not set to zero after some interval than the number of comparisons storing at the receiver increases in higher rate which creates overhead and also utilize more energy.

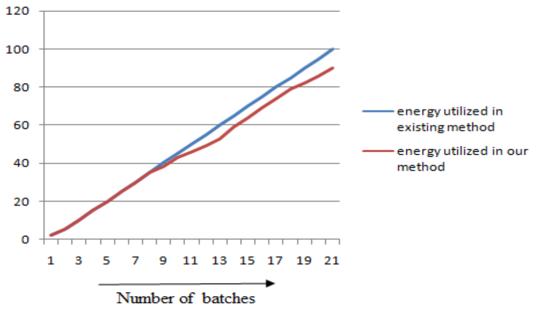

Figure 4: Energy utilized for packet comparison

Figure 4. Shows energy utilized for comparing the packet sequence number in the existing method to process 20 batches each batch containing 05 packets of 1024 bytes, contains uniform reserved bytes for storing packet identification throughout the multicast transmission. In our proposed method for comparison from 1 to 8 batches (each batch containing 05 packets of 1024 bytes) energy utilized is equal to that of the existing method. For batch number 9 to batch number 21 energy utilized is less when compared to existing methods because the reserved bytes for storing is updated (either two or four) than the sequence number comparisons also reduced. The sequence number is calculated from new lower values. The energy utilized in proposed method is

slightly lesser than that of the energy utilized in the existing method, but this small difference in the utilization of energy impacts on the delivery of service in mobile receivers.

4. CONCLUSION

Results in our proposed method shows Duplicate packet detection at the signature header during signature verification using the sequence number method utilizes less comparison and time when compared to existing methods for duplicate packet detection and verification. It is due to resetting the sequence numbers of packets, when the receiver requests for updating the reserved bytes for storing the sequence number. Modifying the reserved bytes for storing sequence number based on the available energy at the receiver utilizes the energy efficiently and processes sequence number in the signature header quickly.

REFERENCES

1. Y. Zhou, X. Zhu, and Y. Fang, **MABS: Multicast Authentication Basedon Batch Signature**, *IEEE Transactions on Mobile Computing*, vol. 9, no. 7, pp. 982-993, July 2010.

2. G. Costa, A. Cuzzocrea, G. Manco, and R. Ortale, ***Data De-duplication: A Review***, Learning Structure and Schemas from Documents. Studies in Computational Intelligence, M. Biba and F. Xhafa, Eds. Springer Berlin Heidelberg, 2011, vol. 375, pp. 385–412.

3. K. A. Farhan, F. Abdel-Fattah, F. Altarawneh and M. Lafi, **Survey Paper on Multicast Routing in Mobile Ad-hoc Networks,** *IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan*, 2019, pp. 449-452.

4. D. Striccoli, G. Piro and G. Boggia, **Multicast and Broadcast Services Over Mobile Networks: A Survey on Standardized Approaches and Scientific Outcomes**, *in IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1020-1063, Second quarter 2019.

5. Shibu K.R , Suji Pramila R **A Novel Secret Key Generation Scheme for MANETs using Traffic Load to Avoid Active Attackers** *International Journal of Emerging Trends in Engineering Research* ***(IJETER)*** Volume 8. No. 5, May 2020.

6. Sangchul Han **Energy-aware EDZL Scheduling of Periodic Tasks on Multicore** *International Journal of Emerging Trends in Engineering Research* ***(IJETER)*** Volume 8. No. 4, April 2020.

7. Vitalii Savchenko , Oleh Ilin , Nikolay Hnidenko , Olga Tkachenko , Oleksander Laptiev , Svitlana Lehominova **Detection of Slow DDoS Attacks based on User's Behavior Forecasting** *International Journal of Emerging Trends in Engineering Research* ***(IJETER)*** Volume 8. No. 5, May 2020.

8. C. Karlof, N. Sastry, Y. Li, A. Perrig, and J.D. Tygar, **Distillation Codes and Applications to DoS Resistant Multicast Authentication**, Proc. 11th Ann. *Network and Distributed System Security Symp. (NDSS '04)*, Feb. 2004.

9. C.A. Gunter, S. Khanna, K. Tan, and S. Venkatesh, **DoS Protection for Reliably Authenticated Broadcast,** *Proc. 11th Ann. Network and Distributed System Security Symp. (NDSS '04)*, Feb.2004.

10. Lysyanskaya, R. Tamassia, and N. Triandopoulos, **Multicast Authentication in Fully Adversarial Networks,** *Proc. IEEE Symp. Security and Privacy (SP '04)*, May 2004.

11. Deng, Shuhua & Gao, Xing & Lu, Zebin & Gao, Xieping. (2017). **Packet Injection Attack and Its Defense in Software-Defined Networks**. *IEEE Transactions on Information Forensics and Security*. PP. 1-1.

12. S. Deng, X. Gao, Z. Lu and X. Gao, **Packet Injection Attack and Its Defense in Software-Defined Networks,** in *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 695-705, March 2018.

13. L. Landmark, Y. Lacharite and L. Lamont, **Multicast Forwarding Using Multiple Gateways and Hash for Duplicate Packet Detection in a Tactical MANET**, MILCOM 2007 **-** *IEEE Military Communications Conference, Orlando, FL, USA, 2007*, pp. 1-7, doi: 10.1109/MILCOM.2007.4454951.

14. Ucar, D. Morato, E. Magaña and M. Izal, **Duplicate detection methodology for IP network traffic analysis,** 2013 *IEEE International Workshop on Measurements & Networking (M&N), Naples*, 2013, pp. 161-166.

15. Golab, Lukasz & DeHaan, David & Demaine, Erik & López-Ortiz, Alejandro & Munro, J.. (2003). ***Identifying frequent items in sliding windows over on-line packet streams***. Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC. 173-178.

16. I Chakeres, **Duplicate Packet Detection for Multicast: Methods, Analysis, and Relative Performance,** 2008 *IEEE Wireless Communications and Networking Conference*, Las Vegas, NV, 2008, pp. 2798-2803.

17. Rahman, Mazedur & Gao, Jerry & Tsai, Wei-Tek. (2013). ***Energy Saving in Mobile Cloud Computing*.*** Proceedings of the IEEE International Conference on Cloud Engineering, IC2E 2013. 285-291. 10.1109/IC2E.2013.37.

18. P. K. Thakur and A. Verma, **Review on Various Techniques of Energy Saving in Mobile Cloud Computing,** 2015 *Fifth International Conference on Advanced Computing & Communication Technologies, Haryana*, 2015, pp. 530-533.

19. P. Jain and A. Suryavanshi, ***Energy efficient Local Route Repair multicast AODV routing schemes in Wireless Ad hoc Network,*** 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, 2014, pp. 1168-1173.

20. A Mohamed Divan Masood, S. K. Muthusunda. **Cryptographic Hashing Method using for Secure and Similarity Detection in Distributed Cloud Data**, *Indonesian Journal of Electrical Engineering and Computer Science* 9 (2018).

21. S. Islam, N. Muslim and J. W. Atwood, **A Survey on Multicasting in Software-Defined Networking,** in *IEEE*

Communications Surveys & Tutorials, vol. 20, no. 1, pp. 355-387, Firstquarter 2018.

22. S. Kim and K. Shin, **A Performance Analysis of MANET Multicast Routing Algorithms with Multiple Sources,** 5th *ACIS International Conference on Software Engineering Research, Management & Applications (SERA 2007),* Busan, 2007, pp. 73-82.

23. J. Chen, F. Yan, D. Li, S. Chen and X. Qiu, **Recovery and Reconstruction of Multicast Tree in Software-Defined Network: High Speed and Low Cost,** in *IEEE Access*, vol. 8, pp. 27188-27201, 2020.

24. Suherman, Suherman & Abdurrahman, Habibi. (2016). **UDP-Lite Enhancement Through Checksum Protection**. March 2017*IOP Conference Series Materials Science and Engineering* 180(1):012146.

25. **Fuzzy logic based proportional integral control of frequency for small**, International Journal of Advanced Trends in Computer Science and Engineering, 2020, volume 9, number 2, pages 1275-1279 Ramaswamy, K. and Dayanand Lal, N. and Parikshith Nayaka, S.K. and Venna, R.C. and Brahmananda, S.H

26. Mr. Parikshith Nayaka S K, Mrs. Shobha Rani, Dr. Dayanand Lal, Dr. M Anand. (2020). **Convert Channel and Information Hiding in TCP/IP** . *International Journal of Control and Automation*, *13*(02), 582 - 591. Retrieved from http://sersc.org/journals/index.php/IJCA/article/view/11199

27. **Effective and Secure Approach for Multi-Keyword Quest Graded over Authenticated Data**, International Journal of Advanced Trends in Computer Science and Engineering, 2020, volume 9, number 2, pages 2278-3091, Shobharani D, Parikshith Nayaka S K, Swasthika Jain T, Dr. Dayanand Lal

28. Jacob, I. Jeena. (2020). **Ensuring Network Security using Secured Privileged Accounts**. International Journal of Emerging Trends in Engineering Research. 8. 1959-1963. 10.30534/ijeter/2020/80852020

29. Dr. Dayanand Lal N, Mrs. Sahana D S, Mrs. Veena R C, Dr. Brahmananda S H, Deepak S Sakkari. (2020). **Image Classification of the Flower Species Identification using Machine Learning**. *International Journal of Advanced Science and Technology*, *29*(05), 995 - 1007. Retrieved from http://sersc.org/journals/index.php/IJAST/article/view/9753