Volume 5, Issue 2, February 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Graph Theoretical Solution for Network Vulnerability Due to ARP Attacks

[1]S. Vidya*, [2]R. Bhaskaran

[1]Fatima College, Madurai, Tamil Nadu, India

[2]School of Mathematics, MK University, Tamil Nadu, India

Abstract— Fastest growing subject within graph theory is the study of domination.. The focus here is to apply it on to reinforce the strength of Computer Networks by identifying and protecting the crucial points of the network and reducing the network vulnerability. Dominating sets play an important role in the reduction of computer network vulnerability, as in communication networks. Therefore the work undertaken here is two pronged, one is to develop an algorithm to generate all possible minimal dominating sets for a given undirected graph of any network topology, and the next step is to arrest the Address Resolution Protocol (ARP) attacks and reduce network vulnerability by monitoring and protecting the pivotal nodes identified from the minimal dominating sets.

Keywords— Address Resolution Protocol, Domination, Minimal Dominating Sets, Network Vulnerability, ARP Attacks

I. INTRODUCTION

Foundations of Graph theory was formed by Euler himself centuries ago. But it is the twentieth century that saw explosive growth in applying graph theory mainly due to the growth in the field of electronics, transportation and information technology. Fastest growing subject within graph theory is the study of domination. This has many and varied applications in fields such as linear algebra and optimization, design and analysis of communication networks, social sciences, bioinformatics, computational complexity and algorithm design. Therefore, it's a field that would interest mathematicians, computer scientists, operation researchers, economists, social scientists, electrical and computer engineers, chemists, system engineers and many others.

Though the mathematical study of dominating sets in graphs began around 1960, the subject has historical roots dating back to 1862 when de Jaenisch studied the problem of determining the minimum number of queens which are necessary to cover or dominate an n*n chessboard [1]. The research in these dominating sets developed further and has left its foot prints in almost every essential fields of human existence. Particularly, dominating sets arise in the study of numerous facility location problems, such as the optimal location of hospitals, fire stations, post offices, mail boxes, schools, stores, radio stations and the like.

The application of dominating sets being so varied, the focus here is to apply it on to reinforce the strength of Computer Networks by identifying and protecting the crucial points of the network and reducing the network vulnerability. Dominating sets play an important role in the reduction of computer network vulnerability, as in communication networks. Therefore the work undertaken here is two pronged, one is to develop an algorithm to generate all possible *minimal dominating sets* for a given undirected graph of any network topology, and the next step is to arrest the ARP attacks and reduce network vulnerability by monitoring and protecting the pivotal nodes identified from the *minimal dominating sets*.

II. TERMINOLOGIES AND LITERATURE SURVEY

A. MINIMAL DOMINATING SETS (MDS)

Definition: In a graph G, a set $S \subseteq V(G)$ is a dominating set if every vertex not in S has a neighbor in S. The domination number $\gamma(G)$ is the minimum size of a dominating set in G [2].

Definition: A set $S \subseteq V$ of vertices in a graph G=(V,E) is called a dominating set if every vertex $v \in V$ is either an element of S or is adjacent to an element of S.

There are several different ways to define a dominating set in a graph, each of which illustrates a different aspect of the concept of domination. A set $S \subseteq V$ of vertices in a graph G=(V,E) is a dominating set if and only if:

1. For every vertex $v \in V-S$, there exists a vertex u such that v is adjacent to u
2. For every vertex $v \in V-S$, $|N(v) \cap S| \geq 1$, that is, every vertex $v \in V-S$ is adjacent to at least one vertex in S.

Definition: A dominating set in a graph G is a set of vertices that dominates every vertex v in G in the following sense: Either v is included in the dominating set or is adjacent to one or more vertices included in the dominating set [3].

In application, the work here is to deal with *minimal dominating sets* with respect to a computer network. A minimal dominating set is a dominating set from which no vertex can be removed without destroying its dominance property. Few observations from these definitions are

1. Any one vertex in a complete graph constitutes a minimal dominating set.
2. Every dominating set contains at least one minimal dominating set.
3. A graph may have many *minimal dominating sets*, and of different sizes.

Let $G = (V, E)$ **be undirected graph with no weights**. A dominating set D of G is a subset of V such that every vertex in $V - D$ is adjacent to at least one vertex in D. A minimal dominating set is a dominating set of minimum number of vertices for G. In the undirected graph represented in Fig.1, the darkened vertices are the members of the minimal dominating sets.

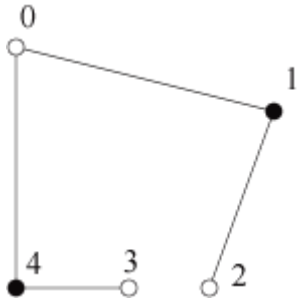


Fig.1 Sample graph showing the MDS vertices as black dots

Some of the dominating sets in Fig.1 are {1,4} and {0,2,3}. The minimal dominating set is based on the cardinality of the dominating set and so the minimal dominating set is {1,4}.

B. LITERATURE SURVEY

A communication network can be considered to be highly vulnerable to disruption if the destruction of a few elements can result in no member's being able to communicate with very many other; Bagga et al [4] based on this idea have suggested the concept of the integrity of a graph - the minimum sum of the orders of a set of vertices being removed and a largest remaining component. They have carried out a survey that includes results on the integrity of specific families of graphs and combinations of graphs, relationships with other parameters, bounds, computational complexity, and some variations on the concept. Xu et al [5] in their paper have proposed a new self-stabilizing distributed algorithm for minimal domination protocol in an arbitrary network graph using the synchronous model. The proposed protocol is designed to stabilize with every possible minimal dominating set of the graph. The paper by Balabhaskar et al [6] aims to provide a detailed survey of existing graph models and algorithms for important problems that arise in different areas of wireless telecommunication. In particular, applications of graph optimization problems such as minimum dominating set, minimum vertex coloring and maximum clique in multihop wireless networks are discussed.

A distributed algorithm for finding approximation minimum connected dominating sets to construct a virtual backbone in the growth-bounded graph for wireless sensor network is presented by Jiansheng Qian et al [7]. This approach consists of three phases. They are, to construct an MIS by network decomposition, to find a minimum dominating set and finally use Marking process and ruling K to optimize the virtual backbone.

The work by Jing et al [8] proposes a simple and efficient distributed algorithm for calculating minimal dominating set in wireless sensor network. This method is said to avoid maintaining the connectivity between backbone hosts. This work proposes a method of calculating minimal dominating set with weight. The nodes are chosen to form a minimal dominating set when the network topology changes. For the host switch on/off operation, the updating algorithm is provided. Johan van Rooij et al [9] have presented algorithms using dynamic programming to count the number of dominating sets. Chaoyi Pang et al [10] have proposed algorithm for dominating sets in directed graphs. In a presentation, Soma Chaudhri [11] has considered both positive and negative results about local, distributed computation. For one, there appears to be a new, feasible, practical algorithm for the approximate MDS problem. On the other, there seems to be worst-case scenarios that demonstrate an absolute lower-bound for computation of minimum vertex cover. A deterministic algorithm requires global information to get a good approximation of minimum vertex cover in all cases. There are also works on finding minimal dominating sets on trees, as in the work of Dorota Br'od [12]. In this work, the author has studied the number of minimal dominating sets in trees.

III. EXPERIMENT AND OBSERVATIONS

The work presented here solves the problem of identifying vital nodes in a computer network, protect them from malicious ARP based attacks and provide proper security for the network using the methods devised to identify and prevent ARP storm [13] and also a subnet based intrusion detection system for tracking down the origin of Man-in-the-middle attack [14]. Considering a computer network, its topology plays a very important role in the way the nodes communicate. Therefore, based on the topology the crucial nodes have to be identified. The life points of the network or the crucial nodes are computers which are life supports for the network to survive. If these machines are under attack or if they do not function to their fullest capacity, then the entire network may be open to the impact of the attack. It could

be in terms of network slowdown or lack of access to resources available on the machine under attack or in other terms a Distributed Denial of Service attack.

To strengthen any network towards vulnerability of ARP attack, some Intrusion Detection System (IDS) or an Intrusion Prevention System (IPS) has to be installed and continuously monitored. The many ready to use IDS and IPS tools that could be installed in the routers, gateways and manageable switches, all have their own shortcomings and none are completely successful. So, it is always best to find localized tailor made solutions to suit the local needs of the subnet. While installing our own IDS or IPS, instead of installing them in all the nodes of the subnet, the idea of installing them in some select points was thought of and the solution is found in the subject of graph theory, specifically using the minimal dominating sets.

As mentioned, this work is carried out in two stages. The first is to create an algorithm for generating the *minimal dominating sets* and the second is to experiment on the network by protecting the pivotal points as indicated by the *minimal dominating sets*.

A. ALGORITHM FOR DETERMINING THE MINIMAL DOMINATING SETS IN AN UNDIRECTED GRAPH

The graph theoretical notation of a computer network is in terms of a weightless undirected graph. The vertices represent the computer nodes of the subnet and the edges represent the physical connection between the nodes. The network topology decides on the kind of connection that exists between the machines.

The software was developed in Microsoft Visual Basic 6 [15]. The same can be done using KBasic [16] which is an open source software. The algorithm was developed to find all possible *minimal dominating sets* available in the input graph. The scalable algorithm is developed for an undirected graph suiting the needs of the selected network.

1) Algorithm for identifying Minimal Dominating Sets

Finding a minimal dominating set had been NP-hard and still is, but even then efficient approximation algorithms do exist. Johan Van Rooij [17] in his work has discussed many algorithms related to the dominating sets. However, these approximation algorithms require a central brain capable of performing global computation, so the question here is whether there is a local, practical algorithm for MDS. The local concept here is each node communicates with its immediate neighbor and so the focus was on the adjacent nodes but not on multihop packet communication involving distance domination. Since there are already many solutions, we aimed at a local, practical solution and evolved our own method.

The algorithm is as follows:

Input: An undirected graph G = (V, E) with n vertices and a set of discarded vertices D∈V if any.
Output: A list containing all minimal dominating sets (MDS) in G, after segregating the redundant lists, dominating sets and non-dominating sets, by exploring all the elements of V.

1: if G is an empty graph then
2: return {Cannot generate a MDS}
3: else
4: loop for n vertices in G
5: if there is a vertex v ∈ D
6: Consider the next vertex in V and loop
7: else
8: if there is a vertex v ∈ G whose degree(v)=0
9: Isolated vertex
10: Add to list MDS(v)
11: else
12: Check degree of v
13: Mark all adjacent vertices of v
14: Create a list P of adjacent vertices
15: Create all such P ∈ M connected components of G starting from vertex v
16: By permutation and combination of vertices from M, generate MDS, or mark the set as redundant or mark the set as non-dominating set.
17: repeat process for all the n vertices
18: return MDS

2) Results

The input graph is presented in Fig.2. The network under consideration for this study is a wired network. In a wired environment there is no possibility of an isolated vertex. But from the graph theory point of view it is possible and hence the input graph in Fig.2 contains isolated vertex. Also, in a computer network, if machines are connected at random, even then the software designed will be able to identify the minimal dominating nodes.

The software generates the results as shown in sample screen shots of Fig.3 and Fig.4. The Fig.3 is a screen shot of the table that contains the list of all dominating sets and the list of all non-dominating sets. Fig.4 shows the minimal dominating set highlighted as red dots in the given input graph. In the sample undirected graph with 9 vertices, one vertex is isolated and others are connected. So the isolated vertex 9 is seen in all the list of dominating sets.

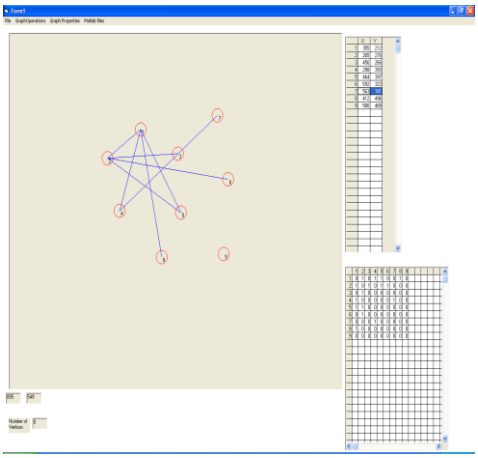
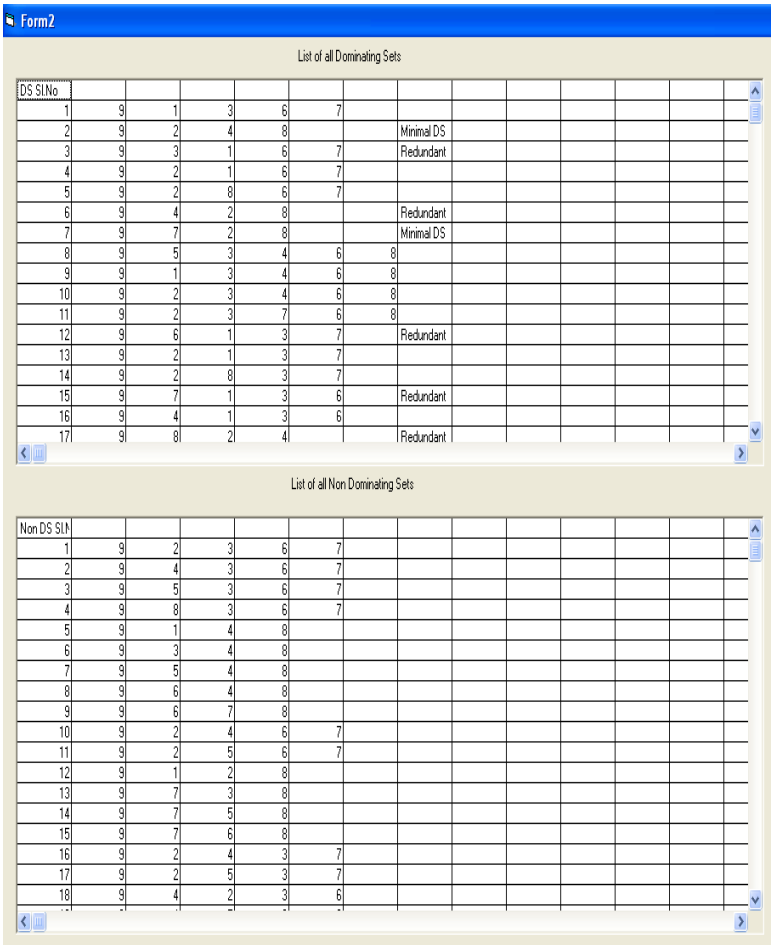


Fig.2 The input graph with 9 vertices



Form2

List of all Dominating Sets

DS Sl.No														
1	9	1	3	6	7									
2	9	2	4	8			Minimal DS							
3	9	3	1	6	7		Redundant							
4	9	2	1	6	7									
5	9	2	8	6	7									
6	9	4	2	8			Redundant							
7	9	7	2	8			Minimal DS							
8	9	5	3	4	6	8								
9	9	1	3	4	6	8								
10	9	2	3	4	6	8								
11	9	2	3	7	6	8								
12	9	6	1	3	7		Redundant							
13	9	2	1	3	7									
14	9	2	8	3	7									
15	9	7	1	3	6		Redundant							
16	9	4	1	3	6									
17	9	8	2	4			Redundant							

List of all Non Dominating Sets

Non DS Sl.N														
1	9	2	3	6	7									
2	9	4	3	6	7									
3	9	5	3	6	7									
4	9	8	3	6	7									
5	9	1	4	8										
6	9	3	4	8										
7	9	5	4	8										
8	9	6	4	8										
9	9	6	7	8										
10	9	2	4	6	7									
11	9	2	5	6	7									
12	9	1	2	8										
13	9	7	3	8										
14	9	7	5	8										
15	9	7	6	8										
16	9	2	4	3	7									
17	9	2	5	3	7									
18	9	4	2	3	6									

Fig.3 The list of Minimal Dominating Sets and Non Dominating Sets

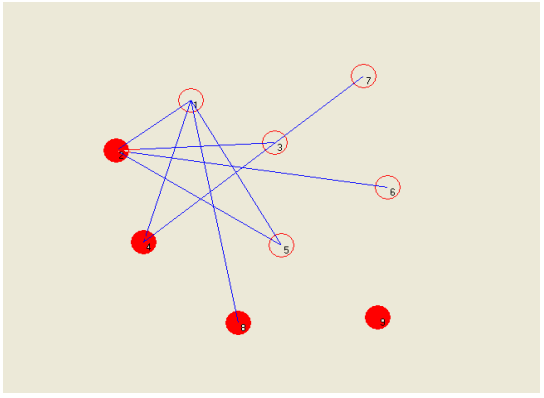


Fig.4 The shaded dots represent the Minimal Dominating Set nodes

To suit a real subnet the algorithm has to be scaled to suit larger subnet. Input graph for the algorithm is drawn based on the topology in which it is needed to identify the machines as pivotal points. The network topology can be bus, star, ring or mesh. The following figures show the nodes in the minimal dominating set with shaded dots in the particular network topology.

Fig.5(a) the Mesh topology being a complete connected graph will have any one node in the *minimal dominating set*. Because, every node is connected to all the other nodes in the graph, each node becomes a dominating node. The ring and bus topology are similar, except for the difference of the ends being connected in ring or being open ended in bus topology.

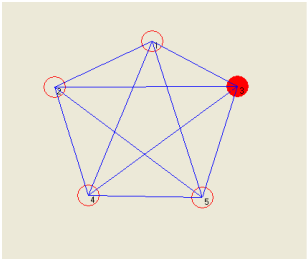


Fig.5 (a) Mesh Topology

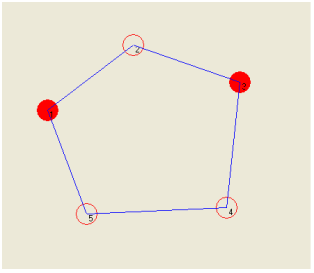


Fig. 5(b) Ring Topology

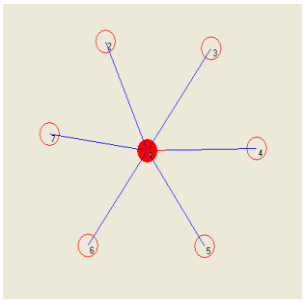


Fig.5(c) Star Topology

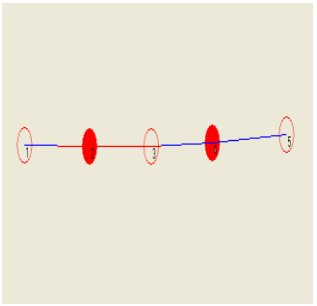


Fig.5(d) Bus Topology

3) Experiment in a test bed network

The experiment was carried out in a laboratory environment. The testing was done on two topologies the star and the bus. Functionally the bus and the ring topologies are the same, except in physical connection of the end nodes being open ended or connected.

As the first step, using the software developed to generate all possible *minimal dominating sets*, given any undirected graph without weights associated with vertices or edges, all the critical nodes of the network were identified. Then the two softwares developed, one for ARP storm detection and prevention [13] and the other for source detection of MITM attack with ARP poisoning [14], were installed in the nodes identified by minimal dominating set as the backbone for the subnet. Now as the network was ready for the test, an ARP storm attack was generated using the Packet Builder tool provided by free edition of Colasoft Capsa software and the MITM attack was made on the node using Cain and Abel. The detection softwares were able to correctly raise alarms and produce alerts for the user to block the attacker/attackers. With the Intruder detection and prevention software running in the network at various vital points, the network was found to be stable on ARP attacks

4) Experimental Observations

This study being done in an academic environment, where, under normal circumstances computers are not often taken in and out of the network. Also the study under consideration is a wired network and so the mobility of the machines is restricted. With this background various observations were made as the experiment was carried out. The observations are listed below:

i. Since the software for detecting and preventing the ARP attacks are designed to deal with the network traffic in promiscuous mode, the software can be installed in any machine and it can identify an attack anywhere in the network. Instead if these softwares are installed in the crucial nodes, then the backbone machine will protect itself and safe guard the entire network from collapsing and also identify an attack anywhere in the network. Thus, giving special protection to the dominating nodes gives added strength to the network.

ii. In this work, we did not stop with identifying the dominating set, but we proceeded to the minimal dominating set. Advantages of identifying the few select nodes of the *minimal dominating sets* are a) Minimal nodes are sufficient enough to monitor the entire network. b) *Minimal dominating sets* help reduce the effort spent on installing the IDS, IPS in all the machines.

iii. If one of the nodes, identified by this algorithm as the critical node is switched off, then it is a serious situation to be under caution. Communication with other machines may not be hindered but the protection for the subnet will be in jeopardy. As mentioned, this work was carried out in an academic network. In this case or rather for any case, the network topology may not be changing every day, so for the machines identified to be crucial, it would be well suited if it could be switched on always. But keeping machines to be switched on always cannot be made

mandatory, hence, if another set of minimal dominating set exists those nodes can be used to supplement the original list of dominating nodes. In case there is only one set of MDS, then network protection may not be guaranteed.

iv. Irrespective of the IDS, IPS being used in the special nodes, identifying a node that maintains connectivity of the network by itself is an essential work to be done. The criticality of the machine is in terms of its physical position in the subnet to maintain the network connectivity and the amount of resources the machine has got to share in the network. If these critical hosts fail to function then it might lead to the collapse of the entire subnet, either by losing the network connectivity or by having a Denial of Service to other hosts in the subnet.

v. If a node is deleted in the original input graph, the algorithm takes care of the situation and produces a new *minimal dominating set*. But in reality, once the vital node is removed from the network, the machine that protects the network is lost. So if there is another set of minimal dominating set they could be evoked, as suggested in the case of a vital node being switched off. If this problem occurs often, then handling this situation does need an administrator. Monitoring the entire network, the moment a crucial node is switched off or removed from the network, software to raise an alarm to the administrator could be set. With the alarm signal, the administrator could be made to send a message across to machines in the next set of minimal dominating set, if there exists one, to install the IDS/IPS tools in their machines. Or if an alternate does not exist then the new MDS should be generated and protective softwares should be installed.

vi. In this work, the minimal dominating set need not be a connected minimal dominating set.

The advantages of using the *minimal dominating set* in this scenario are:

- The network can be maintained in a stable state and total collapse of the network can always be averted just by protecting the pivotal nodes.
- The effort and cost spent on installing an Intrusion Detection System or Intrusion Prevention System on all the machines can be avoided by installing major detection and prevention measures only on the vital nodes.

IV. CONCLUSION

The *minimal dominating set* helps to identify the vital nodes of the network, whose protection ensures safety of the entire network. If any of the nodes in the network fails, that will not affect the entire network, but if the specifically identified machines in the *minimal dominating set* fails to function that will affect all machines directly connected to this machine, one complete part of the network might get affected. This might affect resource sharing and this circumstance might lead to a Denial of Service. Applying the *minimal dominating set* was found to be a simple way of protecting the entire network from collapsing.

REFERENCES

[1] Teresa W.Haynes, Stephen T.Hedetniemi, Peter J.Slater, *Fundamental of Domination in Graphs*, Marcel Dekker Inc. 1998.

[2] Douglas B. West, *Introduction to Graph Theory*, 2nd edition, Pearson Education Asia, 2002, p-116.

[3] Narsingh Deo, *Graph Theory with application to engineering and computer science*, Prentice Hall of India Pvt Ltd., 2004, p-172.

[4] K.S. Bagga, L.W. Beineke, W.D. Goddard, M.J. Lipman and R.E. Pippert, *A survey of integrity, Discrete Applied Mathematics*, 37/38 (1992) 13-28.

[5] Z. Xu, S. T. Hedetniemi, W. Goddard, and P. K. Srimani, *A Synchronous Self-Stabilizing Minimal Domination Protocol in an Arbitrary Network Graph*, Proceedings of the 5th International Workshop on Distributed Computing (IWDC), 27-30 December 2003, LNCS 2918, pp 26-32.

[6] Balabhaskar Balasundaram and Sergiy Butenko *Graph Domination, Coloring and Cliques in Telecommunications*, Department of Industrial Engineering, Texas A&M University, USA.

[7] Jiansheng Qian, Yanjing Sun, *Construction of distributed connected dominating sets in growth-bounded graphs*, Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on 3-5 June 2008, pp 1430 – 1434, 978-1-4244-1717-9 ,DOI: 10.1109/ICIEA.2008.4582755.

[8] Jing Zhang and Chunfu Jia, *Calculation of minimal dominating set in wireless sensor network with host switch-on/off*, Transactions of Tianjin University, Volume 16, No.4, 279-283, DOI: 10.1007/s12209-010-1313-6.

[9] Johan van Rooij, Jesper Nederlof, Thomas van Dijk, *Counting the Number of Dominating Sets With applications in computing the minimum dominating set*, 2009.

[10] Chaoyi Pang, Rui Zhang, Qing Zhang, Junhu Wang, *Dominating Sets in Directed Graphs*, http://ww2.cs.mu.oz.au/~rui/publication/ISci_Dominatingset.pdf.

[11] SomaChaudri, *Discrete Algorithms for Mobile and Wireless Networks,* April 2007, http://www.cs.iastate.edu/~chaudhur/cs611/Sp07/notes/lec22.pdf.

[12] Dorota Br´od, *On the Number of Minimal Dominating Sets in Some Classes of Trees*, Int. J. Contemp. Math. Sciences, Vol. 6, 2011, no. 11, 503 – 506.

[13] S.Vidya, R.Bhaskaran, *"ARP Storm Detection and Prevention Measures*", International Journal of Computer Science Issues, Vol.8, Issue 2, March 2011, pp-456-460, ISSN(Online):1694-0814.

[14] S.Vidya, R.Bhaskaran, "*A subnet based Intrusion detection techniques for tracking down the origin of the Man-in-the-Middle attack*" - International Journal of Computer Science Issues, Vol.8, Issue 5, September 2011, pp-173-179, ISSN(Online):1694-0814.

[15] Microsoft msdn, *Visual Basic 6*, http://msdn.microsoft.com/en-us/vbasic/ms788229, 2011.

[16] Bernd Noetscher's KBasic Software, http://www.kbasic.com/, 2010.

[17] Johan M. M. van Rooij, "*Exact Exponential-Time Algorithms*", ISBN: 978-90-8891-293-1, Uitgeverij BOXPress, Oisterwijk, 2011

