IJCSN - International Journal of Computer Science and Network, Volume 6, Issue 6, December 2017
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

Mobile Agent for Monitoring and Evaluation of Security Applications in a Network Environment

[1] Amosa Babalola; [2] Onyeka Ndidi; [3] Olaniyi Busayo; [4] Babafemi Olusola

[1]Department of Computer Science, San Juan de la Cruz University, San Jose, Costa Rica

[2,3,4] Department of Computer Science, Federal Polytechnic, Ede. Nigeria

Abstract - Mobile agent paradigm derives from two basic disciplines –artificial intelligence from where the concept of an agent originated and distributed systems that define the notion of code mobility having found applications in several areas. However, the implementation of an agent-based system can be through any client/server technology, it is different from classical client/server systems because there is no clear distinction between a client and a server. The mobile agent technology offers several unique capabilities to address the challenges in this area. The objective of this research is to; design a Mobile Agent System for monitoring and evaluating security applications in a network environment and develop a mobile agent capable of increasing the performance of the mobile agent by reducing the size of the agent and also to develop a mobile agent that will be acceptable to all hosts.

Keywords - *Mobile Agent, Performance, Computer Network, Code Mobility, Security Monitoring.*

1. Introduction

The Mobile agents are software abstractions that can migrate across the network, suspend action there and migrate from there to another node, and resume operations from where they left off on the previous node. It has the propensity of communicating in agent language. In concept, a mobile agent can move its complete virtual machine from one host to another. While, on a particular host, the mobile agent can carry out some tasks, and afterward, it can migrate to another host to continue operation from where it stopped action on the previous host. As an executing program, a mobile agent is made up of code, data and execution state and is embedded with some intelligence and the ability to autonomously migrate across the network. As it moves, the mobile agent can represent its owner in various tasks and satisfy its requests. It can communicate with the host as well as other agents on the host; it can sense its environment and carry out a set of activities to attain its mission and achieve its goals [1]. The mobile agent paradigm has become topical since the mid-1990s. It can be deployed in many complex applications such as the internet, Mobile Data Computing, Electronic Commerce, Manufacturing and Scientific Computing [2]. Mobile agent model derives from two fundamental disciplines –artificial intelligence from where the idea of an agent emanates and distributed systems that define the concept of code mobility having found applications in several areas [3].

Managing and monitoring applications in the networks with hundreds of computers have become a challenging and tedious task for today's system administrators. A general computing infrastructure in a medium to a large organization with many nodes, possibly of different kinds, organized into multiple local-area networks and administrative domains. Administration functions require periodic upgrading of software and monitoring application in a network as well monitoring of activities of users at various nodes to defend against potential attacks by miscreants. The complexity of monitoring large regulatory networks— with different kinds of hardware/software components frequently added to the environment or some existing ones periodically upgraded — require new approaches to building network applications, monitoring of protocols and functions. The mobile agent technology offers several unique capabilities to address the challenges in this area.

Many research works have been carried out to measure the throughput/bandwidth of a network [4], [5], [6], [7]. A notable effort to explore through the experiment of significant value of throughput for network assessment and evaluation has been presented in [8]. As presented in [9], a study on distributed network monitoring using mobile agent paradigms is motivated by the fact that network monitoring and management have become necessary due to the proliferation of computers and the rapid growth of the Internet.

An experimental study of bandwidth management in a computer network environment is presented in [10]. Furthermore in [1], [3] mobile agent system for monitoring and evaluation of activities of users in a computer network environment was designed.

In this research, we opt to follow-up on the work of [1] and extend the work to other areas of Monitoring and Evaluation in a network environment; that is the development of a Mobile Agent for Monitoring and Evaluation of Security Applications in a Network Environment.

The goal of this research is the application of a mobile agent system for monitoring and evaluation of security applications in a computer network. The Specific objectives are to design a Mobile Agent System: for monitoring and evaluating security applications in a network environment and capable of increasing the performance of the mobile agent by reducing the size of the agent, which will be acceptable to all hosts.

2. Review of Related Literature

In this section, reviewed literature related to Mobile Agent and Network management and security are as follows: Mobile Agents in [12],[13],[14],[15]. Network bandwidth in [16,[17],[18]. Security Management in a Network is presented in [19], [20], [21], [22] and [23]. Characteristics of Mobile Agents are as presented in [24], [25], [26], [27], [28], [29], and [30]. Network monitoring systems in [31], [32]. Network management and monitoring in [33], [34], [35].

3. Architecture of the Mobile Agent System

The architecture of a mobile system adopted in this research consists of two types of agents namely: the static agents, otherwise referred to as Server agents, and the mobile agent referred to as Agent Monitor, together with their underlying software and hardware infrastructure. On the other hand, the architecture of the mobile agent system can also be categorized as comprising of backend and frontend engines. The backend engine comprises of the server machine and nodes that are considered to be static. The frontend is the software-based interface, which creates the environment for creating and launching the mobile agent and is dynamic in nature. The architecture of the mobile agent is as conceptualized in (Figure 1). The platform for the take-off of the Agent Monitor at the server host and the platform for its landing at the target workstations are their respective operating systems. At the server host, the Agent Monitor is created and equipped with the code, data, and other necessary parameters and dispatched to the target workstations in the network. The Agent Monitor then navigate autonomously through the network from the server end and interact with the host operating system of the target workstations, and it's utility programs as it processes the desired information. The Agent Monitor moves from one workstation to another while carrying along intermediate results. The results obtained by the mobile agent after successful visits to a set of target workstations are transferred to the server, which is displayed on its screen or printed out for the purpose of external analysis, interpretation, policy formulation and decision making by the Network System Administrator. The design of the Agent Monitor has three major components namely: Inputs: the input to the Agent Monitor is the target machine identity, Agent Monitor, and Output report. In the monitoring of security applications, there are two main issues to resolve. One, a system that has to monitor the Security Applications on workstations in the network, and two, a system that has to report back to wherever the request is made. Hence, we have the static agents (server agents) and the mobile agent (agent monitor). The two agents are fussed to design the system.

4. The Mathematical Model

The security application on any computer system is of immeasurable importance. The focus of this research is security application in the computer network Environment. The safety of data and hardware relies on its performance. When a computer is started, the operating system will run a number of processes, security Applications inclusive it's to ascertain the security level of the machine as far as malicious items are concerned. In LAN environment, the issue becomes more problematic. But, these applications may face some problems in the performance. The problems may relate to the agent size. The mobile agent size may increase during a journey. Some hosts may reject to receive the mobile agent in case it has a large size. Also, a mobility of the agent consumes more time and increases the network traffic. This model is an approach that reduces the mobile agent size during its journey. As a result of reducing the agent size, the performance will be improved, and the agent will be more accepted by the hosts. The main idea behind of this approach is to remove some unwanted parts from the agent load (*L*).

As presented in [36], if the MA starts from home, the home node is represented by B_h and the migration can be represented as:

$I = \{B_h, +, n_1, n_2, {}_3,\ldots\ldots n_m\}$……………..…………….(1)

where B_h is the home node and n_1 ($I \geq 1$) is the i^{th} node to be visited in the network. We define I_j as the migration of MA and when the number of nodes to be visited is large (m), the number of MA needed to visit the network is denoted by N Assume that the length of the state information (bytes) is B_s and this is constant throughout the agent lifespan. Then, initial load B_h of MA from home to the first target node is calculated as:

IJCSN - International Journal of Computer Science and Network, Volume 6, Issue 6, December 2017
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

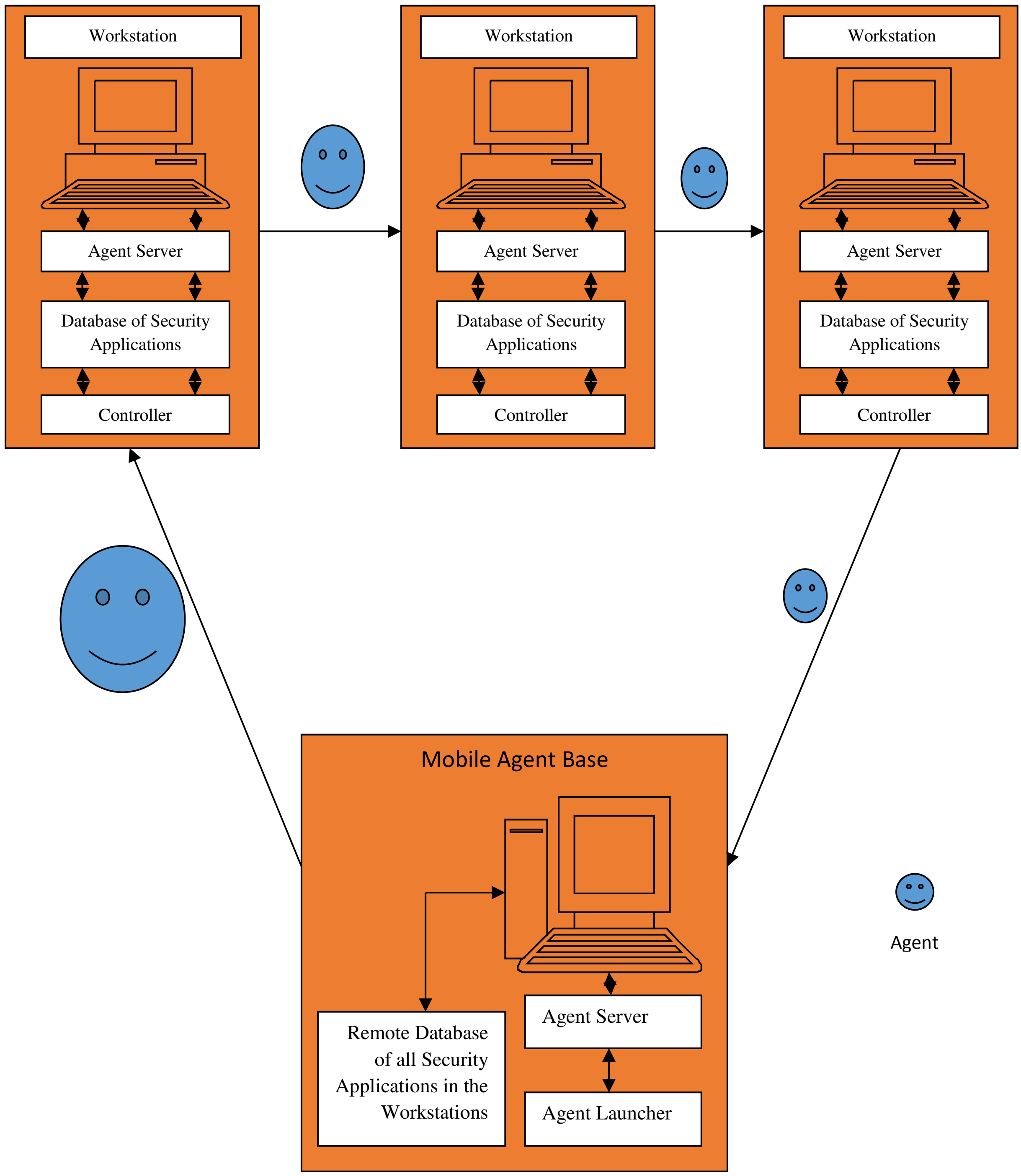

Fig. 1 Architecture of the Mobile Agent

$$B_h = B_c + d_h + B_s \qquad (2)$$

A schematic configuration of the basic structure of a node that has a database of the security application, a controller and an output MA of reduced size is illustrated in (Figure 2).

The load after collecting information on security applications on the first target and the removal of its unwanted contents by the controller attached to the node through regression, whereby reducing its load is B_{h1}. And

the new load B_{h1} of MA from home to the first target node is;

$$B_{h1} = B_c + d_{h1} + B_s \qquad (3)$$

The load of Migration to the next node is;

IJCSN - International Journal of Computer Science and Network, Volume 6, Issue 6, December 2017
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

$$B_{h2} = B_c + d_{h2} + B_s \ldots\ldots\ldots\ldots (4)$$

With this model, the load at each node is further reduced because of the effect of the controller, hence a migration from n_k to $n_{k+1 \text{ with } K} = 1, 2 \ldots,\ldots\ldots\ldots\ldots\ldots$m-1 has a network load of:

$$B_{m1} = B_c + d_{m1} + B_s + \sum_{k=1}^{m-1} d_k \quad \ldots\ldots.. (5)$$

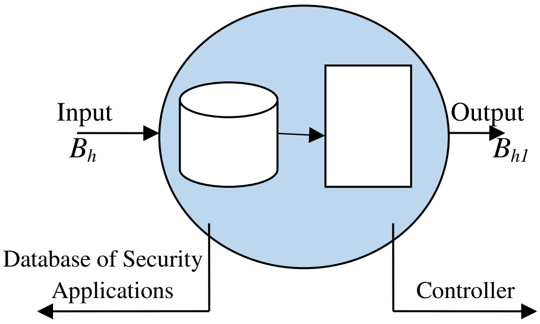

Fig. 2 Schematic configuration of the basic structure of a node with Controller

The load when the Agent returns home (Mobile Agent base) is given by:

$$B_f = d_h + B_s + \sum_{k=1}^{m} d_k \quad \ldots\ldots\ldots\ldots\ldots\ldots (6)$$

The sizes of the MA load in the visited nodes are represented by $B_{h1} \leq B_{h2} \leq B_{h3} \ldots.. \leq B_{hm} \ldots\ldots\ldots\ldots$(7)
Let the load of the output of the first node be denoted as $B_{h1,}$ which is smaller than the load of the Second output B_{h2}; B_{h2} is smaller than the load of the third output
B_{h3} and $B_{h3} \leq$ the load of the last Node B_{hm}

4.1 Performance Tool of the Mobile Agent

The main goal of this model is to increase the performance of the mobile agent. After the mobile Agent completes a part of its journey; this model can reduce a mobile agent size by removing some unwanted parts from its body. Consequent on when a mobile agent data is reduced, automatically the mobility of the agent consumes less time. Also, the mobile agent size will make it acceptable to all the hosts. As defined earlier, the mobile agent comprises of several components. These components represent tasks at each place. After the agent completes a part of its tasks during a journey, some of its components are not useful for the rest of its journey [37]. Therefore, these components are overhead to the mobile agent.

4.2 Migration Process of the Model

Mobility allows the transfer or migration of a mobile agent to another host, as well as the resumption of execution at the new host [38], [39]. In this Model, the Agent is launched (serialized) from the Mobile Agent Base to the next available node, when it reached the node, the agent will be deserialised and collects information about security applications on the node, the agent will move to the attached controller where all its unwanted items will be removed. This process will reduce the size of the agent.

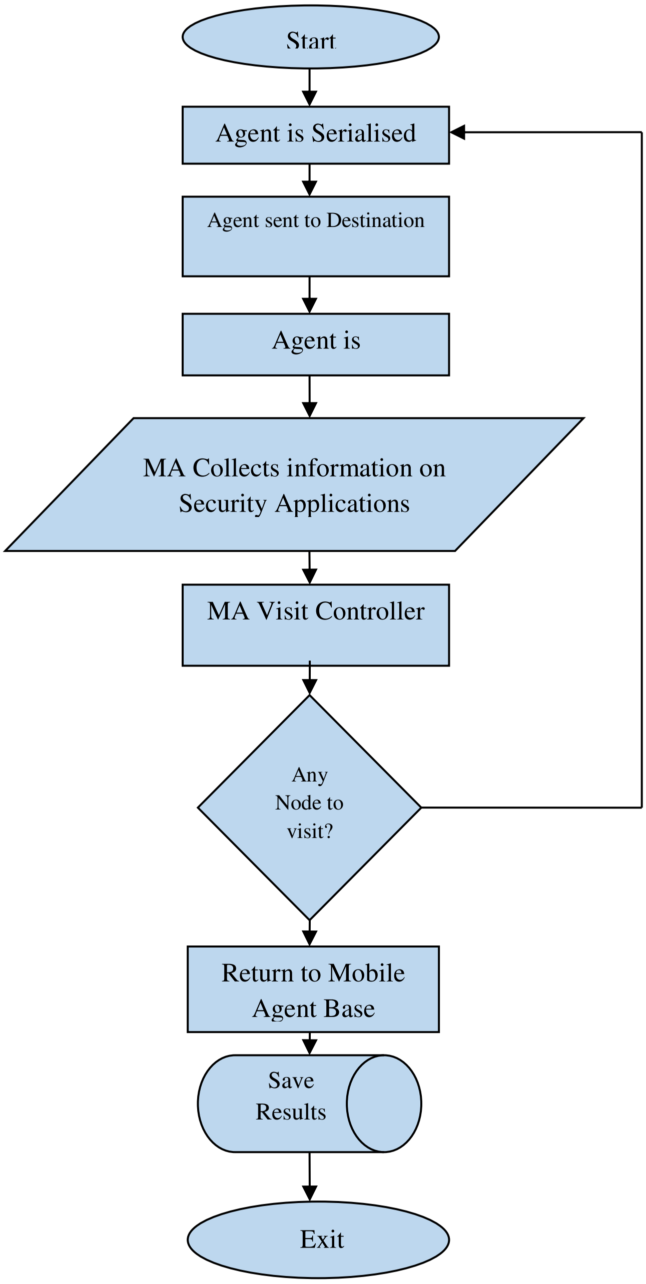

Fig. 3 Mobile Agent Migration Process

The agent will proceed to the next node collects the same information then passed through the controller attached to it. The controller will also remove its unwanted contents to reduce the size of the agent. This process will continue till it reaches the last node. At the completion of its journey, the size of the agent will become smaller. With the reduced size, the performance of the mobile agent will increase. Also, the agent will be able to visit any node because of its size. The flow of the migration Process of the model presented in (Figure 3).

5. Implementation of the Model

The local area network (LAN) environment of the Federal Polytechnic, Ede was used for the implementation of the model. The LAN was tested without the model (non-protected) and with the model (protected). For experimentation of the model 15 nodes were used and the Mobile Agent base (Home node). WINDOW NT is used as the operating platform for the implementation of the mobile agent system and Microsoft access is used as the backend software for the implementation of the mobile agent. Front-end Software used is C-SHARP. However, the choice of C-SHARP (C#) is presented in [40], [41], [42], and [43]. Several choices are available for mobile Agent mobility software which includes: Java, Telescript, Obliq, Agent Tcl, and C-sharp (C#). For the purpose of this research work, C-sharp is adopted as the mobility software. Its features are as presented in [44], [45], [46] and [47].

5.1 Performance Evaluation of the Model

The implementation of this model is in a protected environment. When the agent visits a node, it will collect the information on security applications and then visit the controller component attached to it. The controller will remove all the unwanted items from the database; this will lead to a reduction in the size of the Mobile Agent. Passing of this process from node to node will give a reduced average load during the migration of the mobile agent in a protected environment than in a non-protected environment. The model was tested in protected and non-protected environments with 15 nodes and the host.

5.1.1 Verification of the model in a non-protected environment

During the first experiment (Exp 1), the agent visits the nodes without the controller. In the second Experiment (Exp.2), the agent, visits the nodes with a controller. The results are as follows: The mobile agent base creates a mobile agent. The agent will visit Node1, Node2, Node3, Node4, Node5,…..Node15 to collect information about security applications in each of them. After Completing the migration, the agent returns to home. In (Table 1) the mobile agent size during the journey is presented while a corresponding graph is in (Figure 4).

5.1.2 Verification of the model in a protected environment

The mobile agent base creates a mobile agent. The agent will visit Node1, Node2, Node3, Node4,Node5,…,Node15 to gather information about security applications in each of them. After Completing the migration, thea gent returns to home. In (Table 2) the mobile agent size during the journey is presented while a corresponding graph is in (Figure 5).

Table 1: Mobile Agent size in a non-protected environment

Nodes	Load
Home	7710
1	7300
2	6885
3	6475
4	6065
5	5650
6	5235
7	4825
8	4410
9	3995
10	3585
11	3170
12	2755
13	2345
14	1940
15	1530
Average	**4616**

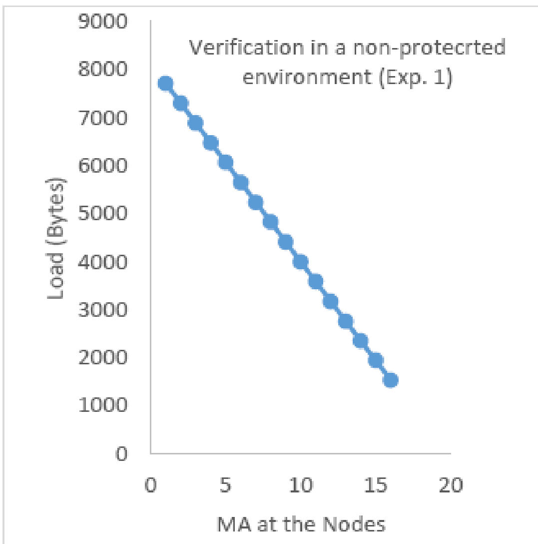

Fig. 4 Verification of the Model in a non-protected Environment

Table 2: Mobile Agent size in a protected environment

Nodes	Load
Home	7710
1	7789
2	7874
3	7954
4	8034
5	8119
6	8204
7	8289
8	8374
9	8485
10	8539
11	8624
12	8709
13	8794
14	8874
15	8954
Average	**8330.1**

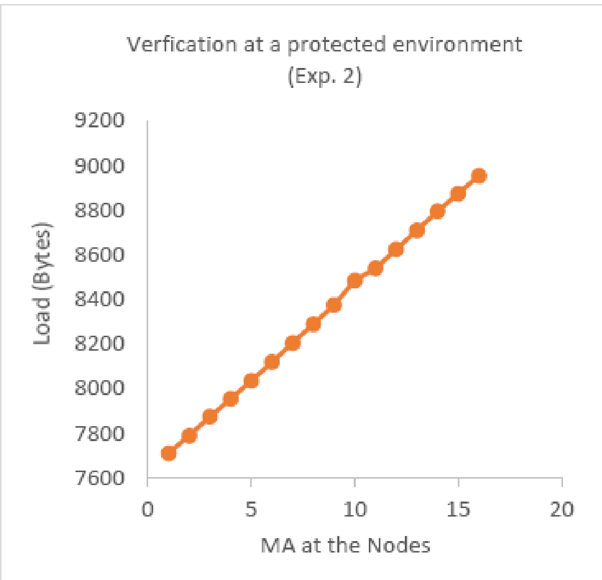

Fig. 5 Verification of the Model in a Protected Environment

6. Analysis of the Results

6.1 Non-protected environment

During its entire migration process the size of the mobile agent in the non-protected environment (Exp.1), increased by 1244 bytes (16%); as presented in (Table 1). Its average size during the journey is 8330.1 bytes. The chart for this experiment is in (Figure 4).

6.2 Protected environment

In the protected environment, the size of the mobile agent (Exp.2), increased by 6180 bytes (80.2%); as presented in (Table 2). The average size of the mobile agent during the migration is 4616 bytes. This increase is because of the inclusion of the controller in the model. The chart for this experiment is in (Figure 5).

6.3 Comparison of Non-protected and Protected Environments

A comparison of the Non-protected and Protected Environments (Figure 6) shows that the performance of mobile agent is better in the protected environment than the non-protected environment.

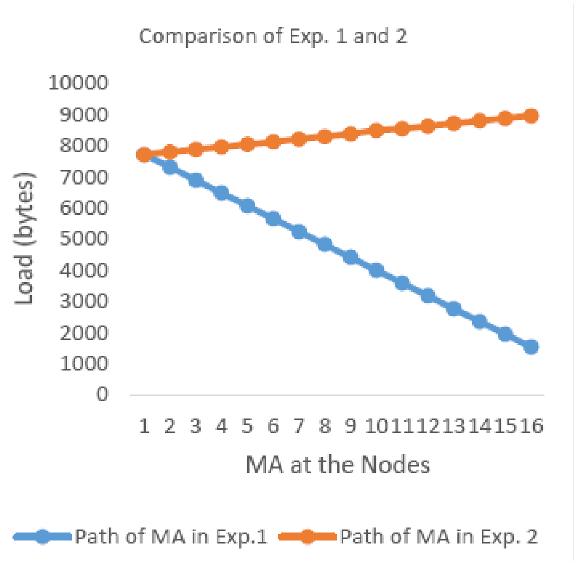

Fig. 6 Comparison of Non-protected and Protected Environments

7. Conclusion

In this research, we have presented how the performance of mobile agent can be increased. The major idea behind the model is to reduce the size of the mobile agent. The model allows the mobile agents to visit all the available places during the migration. During migration, some nodes may refuse the mobile agents because their sizes are large and not acceptable to store them. The model can help in this situation. Two experiments have been performed and according to the result, the model proved its efficiency in reducing the mobile agent size.

The process of monitoring security application on servers and workstations in a network is one of the tedious tasks of the network administrator. A software agent can be dispatched from a server machine to any workstation on the network to monitor security applications available without the network administrator physically moving from one system to another. The mobile agent system for monitoring security application in a network environment is utility software that would enable the network administrator to monitor the security application on all machines in the network.

An architecture comprising of backend engine (server and workstations) and frontend engine (mobile agent) which serves as the software-based interface was proposed. The platforms for the takeoff of the mobile agent at the source and its landing at target destinations are the respective operating system. A mathematical model proposed enables the agent to migrate to a new node carrying all its code, data, and information on security applications passing through the controller.

Analysis from the comparison shows that the mobile agent model performs better in a protected environment than the non-protected environment. The scope of this research is limited in the sense that, it has concentrated on network software tools monitoring an aspect of network management system using a mobile agent. However, one of the limitations of the research work is that it is only one out of the five identified components forming network management system has been studied. Only the security application availability section of the performance management has been implemented.
This model can be implemented in more existing mobile agent systems using the assumptions for their consideration. Also, more research is still necessary for the agent-based implementation of the other network management modules that includes; Fault-tolerant management, Accounting, management, and Configuration management.

References

[1] S. A. Arekete, “Development of Mobile Agent for Monitoring and Evaluation of User Activities in a Network Environment”, Ph.D. thesis, Department of Computer Science, Federal University of Technology, Akure, Nigeria, 2013.

[2] D. Milojicic, “Agent Systems and Applications”. IEEE Concurrency. Vol. 8, Issue 2, 2000, pp. 22-23.

[3] S. A. Arekete, O. C. Akinyokun, O. Olabode, and B. K. Alese, “Design of a Mobile Agent for Monitoring Activities of Users”. Computer Engineering and Intelligent Systems. ISSN 2222-1719, Vol. 4, No 3, 2013, pp. 12-24.

[4] L. Allan, and F. Karen, Network Management: A Practical Perspective, Addison Wesley, 1993.

[5] B. Jean-Chrysostome, “Characterizing End-to-End Delay and Packet Loss in the Internet,” Journal of High-speed Networks. Available at http://ils.unc.edu/dempsey/186500/bolot.pdf, 1993.

[6] K. Lai, and M. Baker, Measuring Link Bandwidths using a Deterministic Model of Packet Delay, ACM SIGCOMM Computer Communication Review, 2000.

[7] H. Saito, and T. Cusho, Design and Implementation of a Network Performance Evaluation System through Client Observation, Meiji University, Japan, 2002.

[8] B. Troy, Network Performance Evaluation Throughput, Computer Science Department, School of Engineering, California Polytechnic State University, California, 1998, pp. 1-20.
Available at;
http://www.ee.calpoly.edu/3comproject/senior-projects/Bennett-Troy.pdf.

[9] F. Kamangar, D. Levine, G.V. Zaruba, and N. Chitturi, “Distributed Network Monitoring Using Mobile Agents Paradigm”, Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA’03), Arabnia H.R. and Mun Y. (Eds.), 2003, pp. 951-957.

[10] A. A, Imianvan, “Development of a mobile agent for evaluating the use of bandwidth in a computer network.” A Ph.D. thesis in the Department of Computer Science, Federal The university of Technology, Akure, Nigeria, 2008.

[11]. O, C. Akinyokun, and A.A. Imianvan, “Mobile Agent System for Computer Network Management.” International Conference on Advances in Engineering and Technology, 2009, pp. 796-808.

[12] M. R. Genesereth, and S. P. Ketchpel, Software Agents. In: Communication of the ACM, 37(7), 1994, pp. 48-53.

[13] V.R. Olga, K. Vladimir, J. Anupam, F. Tim, and Y. Yelena, "Agents2go: An Infrastructure for Location-Dependent Service Discovery in the Mobile Electronic Commerce Environment": In Proceedings, ACM Mobile Commerce Workshop, 2001.

[14] K. Lalana, "A Policy-Based Approach to Governing Autonomous Behavior in Distributed Environments": Ph.D. thesis, University of Maryland Baltimore, 2004.

[15] C. Qiang, and A. Marshall, “Network Management Performance Analysis and Scalability Tests: SNMP vs. COBRA”. IEEE/IFIP Network Operations and Management Symposium, NOMS, South Korea, 2004.

[16] S. Yashpal, G. Kapil, and S. Niranjan, “Dimensions and Issues of Mobile Agent Technology”. International Journal of Artificial Intelligence & Applications. Vol. 3, No 5, 2012, pp.51-61.

[17] C. Lin, and V. Varadharajan, “Mobile trust: a trust enhanced security architecture for Mobile Agent

Systems". International Journal of Information Security. Vol. 9, Issue 3, 2010, pp.153-178.

[18] R. Hassan, and A. Hair, "Towards a New security Architecture of Mobile Agents". International Journal of Soft Computing and Engineering. Vol. 3, Issue 6, 2014, pp. 55-60.

[19] R. Heady, G. Luger, A. Maccabe, and M. Servilla, The architecture of a network Level Intrusion detection system. Technical Report, University of New Mexico, Department of Computer Science, 1990.

[20] T. Zhixin, "A mobile Agent-Based Server System for Server Resources Monitoring". Cybernetics and Information Technologies Vol. 13, No4, 2013, pp. 104-117.

[21] L. S. Zeng, and H. Wang, "Generating Automatic Fuzzy System from Relational Database System for Estimating Null values". Cybernetics and Systems. 40(6), 2009, pp.528-548.

[22] L. Yucheng, and L. Yubin, "A Monitoring System Design Program Based on B/S Mode". Proceedings of 1st International Joint Conference on Intelligent Computation Technology and Automation, China, Changsha, 2010, pp.184-187.

[23] A, K. Sharma, A. Mishra, and V. Singh, "An Intelligent Mobile-Agent Based Scalable Network Management Architecture for Large-Scale Enterprise System". Int., Journal of Computer Networks & Communications, Vol. 4, No 1, 2012, pp.79-95.

[24] M. Breugst, and T. Magedanz, "On the usage of Mobile Agent Platforms in Telecommunication Environments", 5th IS &N Conference, Antwerpen, Belgium, 1998, pp. 25-28.

[25] P. Morreale, "Agents on the move, Spectrum", IEEE, Vol. 35, Issue 4, 1998, pp. 34 –41.

[26] OMG, Agent technology, Green Paper, Technical Report ec#2000-03-01, Object Management Group, March 2000. URL: http://www.jamesodell.com/ec2000-08-01.pdf.

[27] O. Brousse, J. Guillot, G. Sassatelli, T. Gil, F. Grize, and M. Robert, "Mobile Computing Framework for Pervasive Adaptive Computing Platforms". International Journal of Distributed Sensor Networks, 2012, pp.1-15.

[28] A. Mihailovic, A. Kousaridas, A. Jaron, P. Pangalos, N. Alonistioti, and H. Aghvami, " Self-Management for Access Points Coverage Optimization and Mobility Agents Configuration in Future Access Networks". Wireless Personal Communications. Vol. 72, Issue 1, 2013, pp.343-374.

[29] R, Sedgewick, and K. Wayne, Algorithms. Fourth edition. Addison-Wesley Professional, Massachusetts, 2011.

[30] G. Ciobanu, and L. Juravle, "Flexible software architecture and language for Mobile Agents". Concurrency & Computation: Practice & Experience. Vol.24, Issue 6, 2012, pp. 559-571.

[31] R. Boutaba, and J. Xiao, Network Management: State of Art. Edited by B.V. Kluwer. Proceedings of the IFIP17th World Computer Congress – TC6 Stream on Communication Systems: The State of the Art. 2002, pp. 127 – 146.

[32] S. William, SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. Addison-Wesley. MA, 1999.

[33] R. Pinheiro, A. Poylisher, and H. Caldwell, "Mobile Agents for Aggregation of Network Management Data". In First International Symposium on Agents and Applications, and Third International Symposium on Mobile Agents, 1999, pp. 130–140.

[34] M. Crosbie, and E. Spafford, Defending a Computer System using Autonomous Agents. Computer Science Technical Reports, Paper 1200, 1995, http://docs.lib.prudue.edu/cstech/1200.

[35] J. Zhang, "A mobile Agent-based Tool Supporting Web Services Testing". Wireless personal communications. Vol. 56, Issue 1, 2011, pp.147-172.

[36] Y. El-Gamal, E. Khalid and S. Magdy, "A Comparative Performance Evaluation Model of Mobile Agent Versus Remote Method Invocation for Information Retrieval." World Academy of Science, Engineering, and Technology, 2007, pp. 286-291.

[37] M.A. Tarig, "Increasing Mobile Agent Performance by using Free Areas Mechanism". Journal of Object Technology, Vol.6, No 4, 2007, pp. 125-140.

[38] O.C. Akinyokun, J.B. Ekuewa, and S.A. Arekete, "Development of Agent-based system for monitoring software resources in a network environment". Artificial Intelligence Research. Vol. 3, No 3, 2014, pp. 62 – 74.

[39] B.M.G. Amosa, A.A. Sobowale, T. Adepoju, M.A. Hammed, and N. Onyeka, "An Agent Based System for Monitoring Loan Defaulters in Commercial Banks". Academic Journal of Science. Vol. 7, No2, 2017, pp. 45-54.

[40] J, Osborn, Deep Inside C#: An Interview with Microsoft Chief Architect Anders Hejlsberg. O'Reilly Media, 2000.

[41] W. Wylie, Why Microsoft's C# isn't. CNET: CBS Interactive, 2002.

[42] M. Torgersen, New features in C# 4.0, Microsoft Press, 2009.

[43] N. Hamilton, The A-Z of Programming Languages: C#". Computerworld, 2008.

[44] A. Derezinska, Specification of mutation operators Specialized for C# code. ICS Res. Paport 2/05 WUT, 2005.

[45] D. Marshall, Programming Microsoft Visual C#: The Language, 2nd Edition. Microsoft Press, 2008.

[46] H. Scott, C# for the Dynamic Keyword – Whirlwind Tour around. Net 4, 2010.

[47] G.A Aderounmu, A. Adekiigbe, and J.S Iyilade, "An Evaluation of Mobile Agent Paradign". The Journal Computer Science and its Application. 2004, pp. 107 – 116.

Amosa Babalola is a Chief Lecturer at the Federal Polytechnic, Ede, Nigeria Holds a Ph.D. in Computer Science, and he is presently at San Juan de la Cruz, University,
San Jose, Costa Rica. His area of specialization includes; Mobile Agents, Health informatics, Databases and Steganography. He has published a number of referred papers at local and international levels.