library@chuka.ac.ke; www.chuka.ac.ke

MOBILE AGENT BASED SYSTEM FOR LISTING FUNDAMENTAL WI-FI PEER-TO-PEER NETWORK DETAILS

Gogo, K. O.[1] Opiyo E.T.[2] and Barasa, P.[3]
[1]Senior Lecturer, Kenya School of Government and County ICT Manager, Tharaka-Nithi County Government
[2]Senior Lecturer, University of Nairobi, Computer Science Department
[3]Lecturer, Kenya School of Government, Embu. Email: kevingogo2002@gmail.com, opiyo@uonbi.ac.ke, peterbarasa@gmail.com

Citation:
Gogo, K. O. Opiyo E. T. and Barasa, P. (2015). Mobile agent based system for listing fundamental wi-fi peer-to-peer network details. Isutsa, D. K. (Ed.). *Proceedings of the First International Research Conference held from 29th to 31st October, 2014 in Chuka University, Chuka, Kenya, 283-290 pp.*

ABSTRACT
In our everyday operations there is need to engage agents to perform some duties on our behalf, hence they are gaining acceptance as a technology and are being used. Most of the networked offices, networked homes, cyber cafe's, learning institutions and other arenas where computers are interconnected on a Wi-Fi network, have peer-to-peer networks. In Wi-Fi peer-to-peer networks, it is difficult to identify the network details of all the network devices connected such as the IP adresses, mac addresses and computer names of all computers connected on Wi-Fi peer-to-peer network at one go, hereby referred to to as fundamental network details. This is possible in a client-server based architecture where the server monitors all the computers on the network. From the above gap, we developed a mobile agent that could be run in any computer on the Wi-Fi peer-to-peer network and it lists these fundamental details of all the computers connected to the Wi-Fi peer-to-peer network. In developing this mobile agent, we used the MaSE agent methodology. The mobile agent was coded, implemented and tested and subjected to various controls which it overcame and managed to return desired fundamental details with 80% accuracy. The agent had the capacity to classify every computer on the network as either intruder or non-intruder based on the list of authorised computers supplied by the user. The agent suffered major limitation such as taking long to learn and return the results, not communicating to the intruders or shutting them down. In future, the agent could be improved to reduce processing time, communicate and shut down intruding computers or deny them network access.
Keywords: *Wi-Fi, Agents, Network, Peer network*

1.0 INTRODUCTION
On our everyday operations there is need to engage agents to perform some duties on our behalf, hence they are gaining acceptance as a technology and are being used. There is a growing need for practical methods for developing agent applications to easen some of our duties. Wooldridge M. (2004) suggests that agents in multi-agents systems setup are concurrent autonomous entities that coordinate and cooperate so as to perform their various tasks, these coordination and cooperation tasks might be achieved through communication, interaction etc.

Mobile agent based system for listing fundamental Wi-Fi peer-to-peer network details is an agent that is responsible for listing details of the computers found on the Wi-Fi P2P network or workgroup. It lists IP adresses, mac addresses and computer names of the computers. There being various categories of computer networks, this agent is best used in a small Wi-Fi WLAN environments such as in the small cybercafes, schools, networked offices, wireless home networks and other small Wi-Fi P2P computer networks. This agent will be usefull to the computer users and the network administrators as it will be able to monitor who are the Wi-Fi P2P network users, hence discouraging the

illegal network users who do piggybacking on anauthorised Wi-Fi networks. This could be further attibuted to the fact that most home computer networks today are Wi-Fi P2P networks.

1.1 Statement of the Problem

The primary goal of this study is to design and build a fundamental network details listing agent using jade platform, then implement, test and evaluate the agent, with a view to return the fundamental Wi-Fi peer-to-peer network details of the devices connected to that network. The agent also classifies the devices connected on the Wi-Fi P2P network as either intruders or non-intruders to the LAN, based on their IP addresses and could be run on any computer on the Wi-Fi P2P network.

1.2 Research objectives

The research objectives of this project are as follows:-
-To design and build a fundamental network details listing agent for Wi-Fi P2P network using the java platform
-To implement, test, evaluate network details listing agent on existing Wi-Fi P2P network configuration

2.0 RELATED TECHNOLOGIES AND LITERATURE REVIEW

2.1 Related technologies

There are quite a number of related technologies in line with our system in the field of computing. Some of the related technologies and solutions include:

2.1.1 Network management protocols; Yang et al. (2004) argues that a set of automated network management tools usually deals with the multi-vendor environment of the typical installation; a network-management system is based on standardized network management protocols and applications which they are designed to achieve. Due to advanced technology, a large number of network management protocols exist to support network and network device management, although majority of them are not readily available to be used as a package on Wi-Fi peer to peer networks for network details listings. Some of these network management protocols which are highly in line with this project include and not limited to:

a. Simple Network Management Protocol (SNMP)
b. Common Management Information Protocol (CMIP)
c. XML-based network management.

2.1.2 Microsoft Windows Management Instrumentation (WMI). The WMI technology is the Microsoft implementation of the Distributed Management Task Force (DMTF) Web-Based Enterprise Management (WBEM) initiative that extends the Common Information Model (CIM) to represent management objects in Windows-based management environments. The Common Information Model, also a DMTF standard, is an extensible data model for logically organizing management objects in a consistent, unified manner in a managed environment (Microsoft 2006). Based on the Common Information Model**,** WBEM is a DMTF initiative and technology that establishes management infrastructure standards and provides a standardized way to access information from various hardware and software management systems in an enterprise environment. WBEM provides a point of integration through which data from management sources can be accessed.

2.1.3 Peer to peer File sharing software systems, some of these file sharing software like Kazaa and Napster rank amongst the most popular software applications enable the exchange of some network information across the network, as well as the music and video files across the networks. Although majority of these file sharing software do not disclose the fundamental network details of the all the devices connected on a Wi-Fi peer to peer networks.

2.2 Literature review

2.2.1 Computer networks and the IP addresses

Opiyo, E. et al. (2006) suggest that computer network speeds have also increased, which has made it possible to exchange information around the world in a much more feasible manner; in view of this we now hear and read about globalization where the world is tending to become one global village. For the computers on the network to communicate to each other there has to be some set of rules which have to be followed to effect communication, these rules are referred to as the Internet Protocol (IP). There are various computer networks namely:- Local area networks (LAN), Metropolitan area networks (MAN), wide area networks (WAN), virtual area networks (VAN), and personal area network (PAN).

El-Rewini H., Abd-El-Barr M. (2005) maintains that IP address is an exclusive number that all information technology devices (printers, routers, modems, etc) use which identifies and allows them the ability to communicate with each other on a computer network. The IP address could be either Static IP or a dynamic IP. The IP addresses are moving from IP version 4 (IPv4) currently used by most network devices to IP version 6 (IPv6), the move is due to more and more computers accessing the internet.

2.2.2 Peer-to-peer / adhoc networks

Odell J., et al. (2000) view that Peer-to-peer connection is a connection where any computer on the network can be a client or a server or both at the same time, i.e. there is no dedicated server to control the clients. P2P networking type is most commonly used computer networks. This network type is very cost effective but supports lesser number of computers in network, about ten to fifteen computers can be connected to each other using P2P networking model without problem, more number of computers often create problems.

2.2.3 Wi-Fi networks

Michiardi, P. and Molva, R. (2003) explains that Wi-Fi is a communication technology that uses the 'Direct-sequence spread spectrum radio technology' and the 'Orthogonal Frequency Division Multiplexing radio technology', Wi-Fi is the trademark used by a trade group known as Wi-Fi Alliance. Wi-Fi networks Operating in peer-to-peer mode allows all wireless devices within range of each other to discover and communicate in ad-hoc fashion without involving central access points including those built in to broadband wireless routers. Below is a figure 1 showing Wi-Fi peer-to-peer network.

Figure 1: Wi-Fi peer-to-peer network. *Source <http://www.wifinotes.com>*

2.2.4 Agents

Nwana H.S. (1996), claims that agents are autonomous programs that can perform services on behalf of the user. Agent designing is task-oriented. Instead of looking at what actors are involved in an operation, you look at what tasks and subtasks the operation consists of. Agents are then created to solve these tasks. Whereas object orientation does not say anything about the actual tasks but rather expects the objects to solve them implicitly, agent orientation concentrates on the tasks at hand and creates actors that can help in solving these tasks. There are various categories of agents as illustrated below: -

2.2.4.1 Network agents

There are a numerous network agents that have been designed and are in existence, but a majority of the agents are biased towards network security, client/server networks among others. Most of the network details listing agents are embedded within the operating systems, but this is merely to list only a few network details such as the IP address of the computer that the operating system is running within as illustrated by Odell J., et al (2000). Network features, such as Network card speed, Network switch speed, etc., does impact the network throughput of these agents.

2.2.4.2 Mobile Agents

Mobile Agents are programs that can migrate from host to host in a network. They should be able to execute on every machine in a network and the agent code should not have to be installed on every machine the agent could visit. Therefore Mobile Agents use mobile code systems like Java and the Java virtual machine where classes can be loaded at runtime over the network. Below is figure 2 showing a mobile agent movement. Michael W. (2002) suggests that mobile agents are seen as a potential threat to systems. If mobile agents were misused they could generate denial of service attacks or even steal data. Most mobile agents are heterogeneous so that they can travel around a heterogeneous network; hence they have all the characteristics of an agent. Mobile agents send not only data, but code, hence more network bandwidth is required. This is often offset or even reduced to lower level than without mobile agents by the fact that data can be processed at various locations in the network.

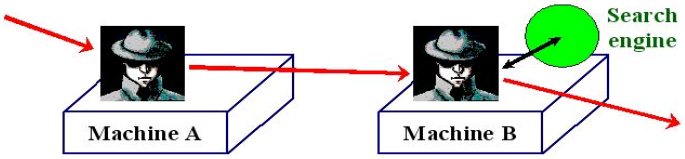


Figure 2: Mobile agent illustration
Source < http://www.limsi.fr/~jps/enseignement/examsma/2004/BHATTI/index.htm#Mobile%20Agents >

2.2.5 Mobile agents programming
As suggested by Tarau, P. et al (1997), there are various programming languages that can be used in implementing mobile agents, as long as the language has the following capabilities: - support for agent migration, support for agent-to-agent communication, support for interaction with local resources, security mechanisms, suitable execution efficiency, language implementation across multiple platforms, and Ease of programming of the tasks mobile agents perform. We used jade programming language in coding our agent, though there are other agent programming languages such as aglets, odyssey, Voyager, telescript, oblique and perl5 among others.

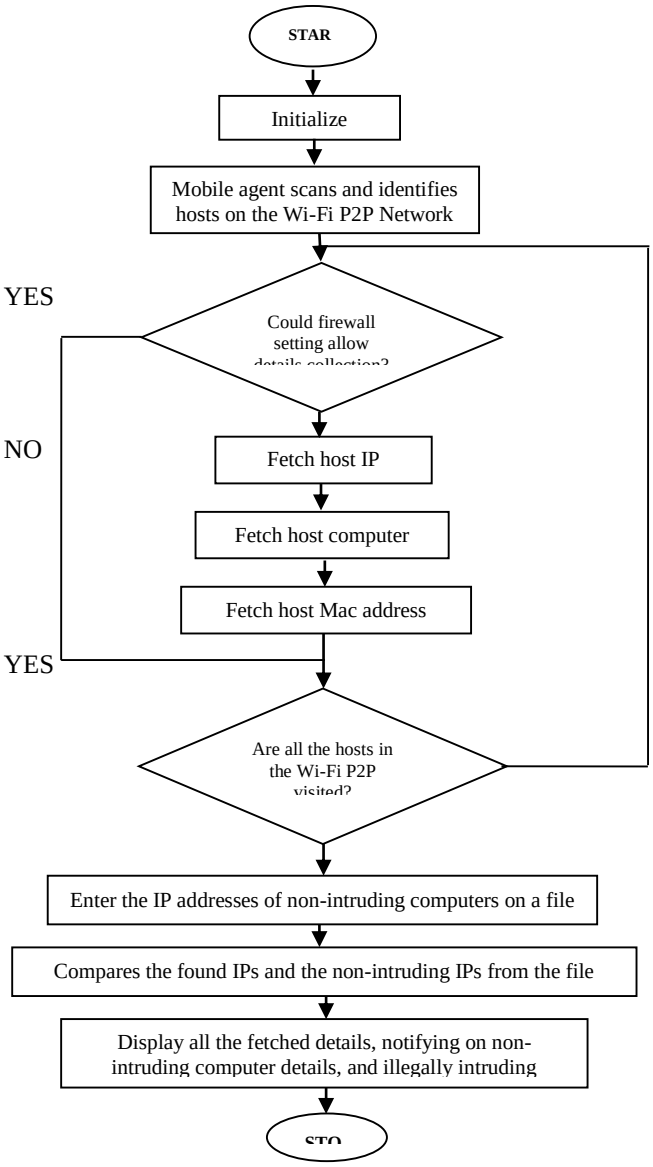
3.0 METHODOLOGY
3.1 Systems design
In system designs as advised by Chauhan (1997), we used ***MaSE*** (multi agent system engineering) methodology. This methodology advocates for two phases namely analysis phase and designing face. In ***analysis phase*** we captured the agent's goals, applying use cases and refine agent's roles. While in ***designing phase*** we created the agents classes, assemble agent's classes then finally design the agent, as shown in table 1 below.

Phase	**Models**
1) Analysis phase	
a) Capturing goals	Goal Hierarchy
b) Applying use cases	Use Cases, Sequence Diagrams
c) Refining roles	Concurrent Tasks, Role Model
2) Design phase	
a) Creating agent classes	Agent class diagrams
b) Constructing conversations	Conversation diagrams
c) Assembling agent classes	Agent Architecture Diagrams
d) System design	Deployment Diagrams

Table 1: MaSE methodology steps

3.2 Experimental and data sources design
We began by setting up a Wi-Fi P2P network. In setting up this network we send the Wi-Fi signals from a wireless router or joined a P2P WLAN hotspot. We then ran our prototype in any randomly sampled computers forming the Wi-Fi P2P network and documented the results. We also added and removed other computers on the network and ran the prototype, then recorded the results obtained. The recorded results constituted the data which we used for the analysis of the mobile agent.
3.3 Data collections procedures, methods and tools
In data collections as argued by Kothari C.R. (2004), in our project we collected the data through experiments and observations methods. The data was recorded at source, immediately they were collected.
3.4 System implementation and testing
In the implementation of the fundamental network details listing agent, first we had to setup a Wi-Fi P2P network. In setting a P2P network, we either used Wi-Fi. Then we installed JAVA, this is because we are using java based framework. We then ran the system by invoking the agent.
3.5 Data analysis and Evaluation
In analysis we printed the screen shots of the results, graphs, tables and pie charts. We analyzed and evaluated the data with regard to real life experience and some sets of controls. The analysis and evaluations was done in a way such that we compared the data that the agent gave under various tests, and the actual information on the ground. Some of these comparisons were; how many fundamental computer network details does the agent return, while actually how many computers are connected to the Wi-Fi P2P network?

4.0 IMPLEMENTATION
4.1 Conceptual framework



4.2 Sample code for analysis agent

This is a sample code for the analysis agent , which analyses the IPs from the network scan agent and the allowed IPs file, and compares them to classify the computers as either intruding or non-intruding.

```
package agents;
import abstracts.AbstractAgent;
import behaviours.ReceiveMessages;

// @author gogo

public class AnalysisAgent extends AbstractAgent{
   @Override
   protected void setup()
   {
      register("Analysis Agent");
      addBehaviour(new ReceiveMessages(this));
   }
}
```

4.3 Sample code for network scanner

This is a sample code that scans the Wi-Fi P2P network and returns the network details that have been identified and forwards them to the analysis agent. Its in this code that one can change the range of IPs to be scanned.

```
/*
 * To change this template, choose Tools | Templates  and open the template in the editor.
 */
package agents;
import abstracts.AbstractAgent;
import behaviours.GetAllIps;

//@author gogo

public class NetworkScanner extends AbstractAgent {
   @Override
   protected void setup() {
      register("Network Scanner");
      int range = 2;  //No of machines scanned by one behaviour
      for (int i = 1; i < 20;) {
         int pass = ((i+range)-1);
            if(pass>254)
               pass=254;
         addBehaviour(new GetAllIps(this, i, pass));
       //System.out.println(i+" - "+((i+range)-1));
         i += range;
      }
   }
}
```

5.0 TESTING AND RESULTS

5.1 Testing the system when connected to a network and no allowed IPs entered

When the system was connected to a Wi-Fi network connecting 4 computers without computer names, it returned the details of the 3 computers connected to the Wi-Fi network, missing one computer. It managed to classify all the IPs of the detected computers as INTRUDING , this is because we did not entered any computer IP address on the allowed IP file and so any detected IP will be classified as INTRUDING. Figure 3 below shows the systems interactive GUI with the results.

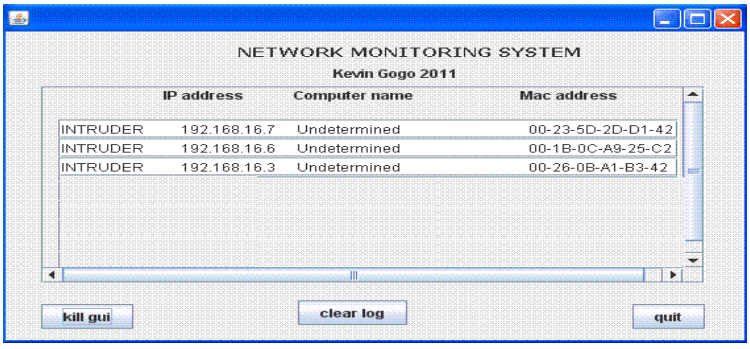


Figure 3: Systems GUI showing results when there is network but with no allowed IPs given

5.2 Testing the system when connected to a network with allowed IPs entered

The system was made to run again this time with allowed IP (41.89.64.60), it categorized the computer with the given IP as allowed as shown in the figure 4 below.

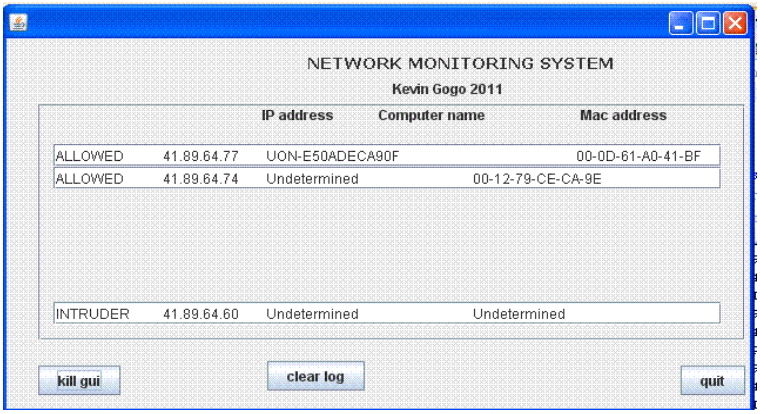


Figure 4: Systems GUI showing results when there is network with allowed IPs entered

5.3 Testing the system when connected to a network with both static and dynamic IPs

The agent was run on one network with 9 computers with some computers having static IP address it identified all the 9 computers and classified them accordingly. It also managed to classify the IPs that are in the allowed file (192.168.1.100) as allowed IPs, as shown in the figure 5 below:-

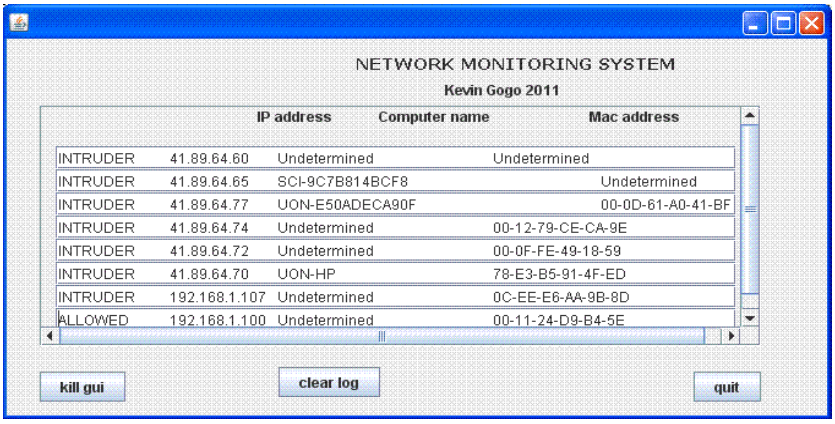


Figure 5 - System GUI showing results when networks changed as system runs

6.0 CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

The main objective of the mobile agents that we have developed was to relay the fundamental details of all the computers connected on the Wi-Fi P2P network. These computer network details are:- computers IP adresses, computer names and the Mac adresses. The agent was developed and tested on various Wi-Fi P2P networks as well as the cabled networks. The mobile agent successfully achieved its intended objectives, but with some slight challenges in others. It worked well on networks which consisted of a few computers, especially with IP adresses following each other sequentially. The mobile agent also classified the computers found as either allowed or intruding, which will help the network provider or user to know who is piggybacking on his/her network.

The main challenge that the mobile agent is facing is time consuming. In essence to scan a single IP address the agent takes about 2 second, which implies that if the IP range is from IP 192.168.1.1 to 192.168.1.20, then the system will take about 2minute 30 seconds inclusive of time for learning its path which is on average 1.75 minutes. In some occassions, especially when the Wi-Fi network was weak, the agent returned a few IP addresses with an accuracy of about 80%.

6.2 Recommendations

The system managed to solve a greater part of the problem which was at hand. We hereby recommend the following further developments on the systems: -.

- Reducing the time it takes the multiagent to go through a P2P WiFi network and return a substancive network detail without a miss
- Detects your internal and external IP addresses.
- Retrieves currently logged-on users, configured user accounts and uptime.
- Supports Wake-On-LAN, remote shutdown and sending network messages.

REFERENCES

Chauhan, D. 1997. JAFMAS: A Java-Based Agent Framework for Multiagent Systems Development and Implementation, master's thesis. Electrical Engineering and Computer Science Dept., University of Cincinnati, Cincinnati.

El-Rewini, H. and Abd-El-Barr, M. 2005. Fundamentals of computer organization and architecture. A John Wiley and Sons, Inc.

Joey, P. 2010. Peer to peer configuration, [online]. available from http://www.wifinotes.com/computer-networks/how-to-configure-network.html [accessed on10th September 2014]

Kothari, C.R. 2004. Research Methodology, 2nd ed., New Delhi: New Age Int. P Ltd.

Michael, W. 2002. An Introduction to Multiagent Systems. John Willey and Sons, Ltd.

Michiardi, P. and Molva, R. 2003 Ad hoc networks security, IEEE Press, Wiley, New York.

Microsoft 2006. Microsoft Data Network knowledge base. Viewed on 12th Sep 2014 <http:// www.microsoft.com>

Nwana H.S. 1996. 'Software agents: An Overview', in Knowledge Engineering Review.

Odell J., Parunak H. and Bauer B. 2000. Extending UML for agents. In Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence.

Opiyo E., E. Ayienga, K. Getao, B. Manderick, Okello-Odongo and A. Nowé 2006. "Computing Research Challenges and Opportunities with Grid Computing". *In*: Ddembe W. and Baryamureeba V. (eds.). Measuring Computing Research Excellence and Vitality. Fountain Publishers, Kampala.

Phillips, M. 2010, Mobile and Ambient Agents in the Internet and in the Electronic Corposants', available from http://www.limsi.fr/~jps/enseignement/examsma/2004/BHATTI/index.htm#Mobile%20Agents, [accessed on 10th March 2011].

Rapaport, T.S. 2002. Wireless Communications Principles and Practice, Prentice Hall.

Tarau, P., Dahl, V. and De Bosschere, K. 1997. High level logic programming tools for remote execution, mobile code and agents. Accessed on 8th May 2011 at http://clement.info.umoncton.ca/html/remote/art.html

Tesfatsion, L. 2001b. Agent-based modeling of evolutionary economic systems. IEEE Transactions on Evolutionary Computation, 5:1-6.

Wooldridge, M. 2004. An Introduction to Multi-Agent Systems, Wiley.

Yang, H., Luo, H., Ye, F., Lu, S. and Zhang, L. 2004. Security in mobile ad hoc networks: Challenges and solutions. In: Proc. IEE Wireless Communication. UCLA, Los Angeles, CA, USA; Volume 11, ISSN: 1536-1284