IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

A Multi-Agent Based System Architecture for Monitoring Nodes in Local Area Networks

John-Otumu Adetokunbo M[1], Ojieabu Clement E[2], Oshoiribhor Emmanuel O[3]

ICT Directorate, Ambrose Alli University, Ekpoma, Nigeria[1]

Department of Electrical/Electronic Engineering, Ambrose Alli University, Ekpoma, Nigeria[2]

Department of Computer Science, Ambrose Alli University, Ekpoma, Nigeria[3]

Abstract: Since it is not easy for network administrators to monitor nodes in a network environment manually due to the physical movement from one node to another; and it is even worst with the trend of the increase in nodes in a network environment due to Web based Service Oriented Applications (SOA) that runs on the networks. Monitoring of nodes in terms of detecting faulty network cables, detecting of nodes that is not supposed to be part of a particular network, and even the shutting down of hundreds of nodes has become a nightmare for most network administrators. The static software solutions and dynamic single task software for network management developed has not really helped matters on the network administrators end. These worries gave birth to this research work, of designing a multi-agent based system to handle different task for the sole benefit of the network administrator and not the network environment. The system is effective and efficient, and it is recommended for practical usage.

Keywords: Multi-agent based system, Mobile agent, Monitoring, Nodes, LANs

1. INTRODUCTION

The term "agent" according to [1] originated from the Greek word "agein" meaning to drive or to lead; but in Computer science, agent denotes a system that is situated in some environment and is capable of autonomous actions. The concept of agents originated in AI (artificial intelligence) and especially in the field of agent and multi-agent technology [2, 3, 4], thus the roots extend back to the 1950s. Software agents are more or less "intelligent" chunks of computer code that are able to perceive and communicate with each other and react to stimuli in order to pursue their goals [2]. According to [5] agents are autonomous decision-making units with diverse characteristics. Autonomous in agents means that they could be asked for the action, but it is up to them whether they will act or not [2].

A multi-agent system is composed of multiple autonomous units called agents that operate in the same environment targeted at realizing a common goal. Multi-agent systems belongs to a wide family of artificial intelligence and are tightly interconnected with other fields like neural networks, genetic algorithm, fuzzy logic and machine learning [2, 6, 7]. Multi-agent based systems are currently being used in wide variety of applications, ranging from comparatively small systems for personal assistance to open, complex, mission critical systems for industrial applications [8].

An example of the application of multi-agent systems includes system diagnostics [9] and network management [10] which needs to be monitored and managed in real time. Multi-agent systems are also used both for the management of distributed networks and for the realization of advanced telecommunication services [11, 1].

A computer network according to [12] is a set of devices (often referred to as nodes) connected by communication links, while the nodes can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network; but [1] defines computer network as a collection of physically separated computers which are connected together primarily to search for, share and exchange computer resources.

The process of administering and monitoring of nodes activities in a computer network is a very tedious task for network/system administrator. Monitoring and evaluation of nodes with a view to resolving problems, and ensuring optimal performance and efficiency normally involves the physical movement of the network administrator from one computer system (node) to another.

1.1 Research Background

Conventionally, the monitoring, searching for resources and administration of nodes in a computer network involves the physical movement of the network administrator from one computer system or node to another [13, 14]. However, nodes in a network environment are increasing very fast due to the appreciable level of the use of information and communication technology (ICT) in well-structured organizations for intranet and Internet based application solutions for services like computer based instruction (CBI) for teaching / learning, computer based testing (CBT) for conducting online and off-line based examinations, and so on.

The manual monitoring of software tools and nodes in a network environment by network administrators can be a

DOI 10.17148/IJARCCE.2016.57157

very huge task and cannot satisfy the requirements of the modern complex network system [1].

For example, monitoring and administration of the computer systems (nodes) in terms of detecting faulty network cables / nodes, detecting anomaly nodes intrusion, and even the shutting down over 250 computer systems after usage within a stipulated time on a daily basis by one or two network administrator in the network environment manually is a very stressful and boring task.

From literatures reviewed, the evolution of software diagnostic packages for troubleshooting nodes in network environment has not really helped much in this regards, since most of the developed diagnostic packages are static in nature [13], and in client/server approach, tasks are also statically defined [15, 16]. Also, the software solutions that are developed to be dynamic for network management using agent technologies for monitoring network environment seems to perform single task only like an Agent-based System for Monitoring Software Resources in a Network Environment [1], Mobile Agent for Evaluating the Use of Bandwidth in a Computer Network [13], Mobile Agent for Monitoring and Evaluation of Activities of Users in a Network Environment [14], Multilevel Monitoring and Detection System [17].

None of these simulated or developed software solutions for network management has been able to harness their research potentials in solving the basic network/system administrator's pains in terms of shutting down large number of computer system and detecting faulty cable connections / nodes amongst others at real-time occurrence by developing and deploying a network management software solution for monitoring/performing multiple functions to the network administrator's advantage. This is the gap in knowledge this research work intends to fill; that is, to integrate and configure the different monitoring task together into one solution platform with short message service (sms) gateway web services integration for reporting the real-time monitoring via mobile phone alert messaging using a multi-agent based system.

[1] Recommends that network administrators will function more efficiently if agent-based network monitoring system is designed to include configuration management, fault detection and security.

The aim of the study is to design a single platform with web services integration for effective real-time monitoring and reporting of nodes activities in a network environment using a hybrid multi-agent based system.

2. DESIGNING THE MULTI-AGENT BASED SYSTEM ARCHITECTURE

System architecture intuitively denotes the high level structures of software. It can be seen as the structures needed to reason about the software system which comprises the software elements, the relationship between them, and the properties of both elements and the relations. The proposed multi-agent based system for monitoring nodes in network environment is made up of five agents. The agents will be able to store a representation of the outer world in their memory and they can make decisions about their present and future actions on the basis. The power of the multi-agent based system stems from the agent's interactions and inter-operation.

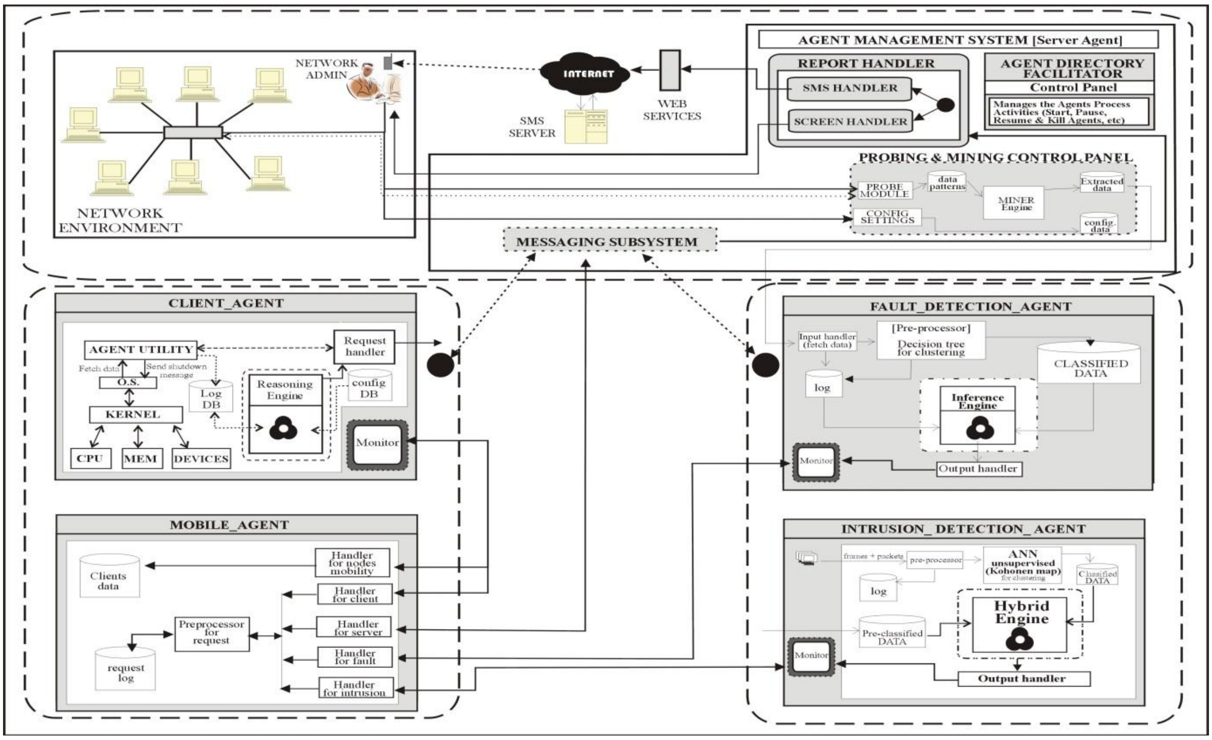

Figure 1 System Architecture for the Multi-agent Based System

Each agent will be able to coordinate its conduct with another agent in order to reach the objective it cannot manage alone; and each agent can also cooperate with another agent if it faces a task that it cannot accomplish. The agent's acceptance or rejection of request is completely dependent on its own objectives or goals.

Figure 1 depicts the system architecture of a multi-agent based system architecture showing the various agents managed and controlled in a single platform referred to as agent management system residing in a network of interconnected nodes.

The model will be further designed having in mind its functionality, as well as performance issue relating to the multi-agent based system and the network environment in which it operates.

2.1Designing of the multi-agent based system

The mathematical notation of the Agent Management System is represented by the three items.

AMS = {MA, S, N}

Where:

MA – a multi-agent system consisting of 5 agents cooperating and communicating in an environment defined by S and N.

S – a set of processing nodes in which the agents perform services

N - a network environment that connects processing nodes and allows communication.

The agents in the multi-agent system MA = (sA, cA, mA, fA, iA)

Where: sA, cA, mA, fA, iA is represented by = (server_agent, client_agent, mobile_agent, fault_agent and intrusion_agent) performs some tasks on behalf of its owner MA. They can also interact and communicate with other agents autonomously to perform some kind of service.

The set of processing nodes is denoted as S = {S_1, S_2, S_3, S_4, …, S_i, … , S_n}. Each node S_i can provide an operating environment for the agents.

A network environment N connecting nodes from S is represented by an undirected graph,

N = (S, E).

E is the set of links, E = {i_1, i_2, i_3, …, i_m. …, i_n} where i_m = {S_1, S_2} which represents a link between nodes S_1 and S_2.

The functionality of the multi-agent based system is defined by a set of elementary services {es} supported by the each agent.

mobile_agent = mA = {es_1, es_2, es_3, $es_{4,}$ es_5}
server_agent = sA = {es_1, es_2, es_3, es_4}
client_agent = cA = {es_1, es_2, es_3, $es_{4,}$ es_5}
fault_agent = fA = {es_1, es_2, es_3, es_4}
intrusion_agent = iA = {es_1, es_2, es_3}

2.2Multi-Agent Based System Communication Scheme

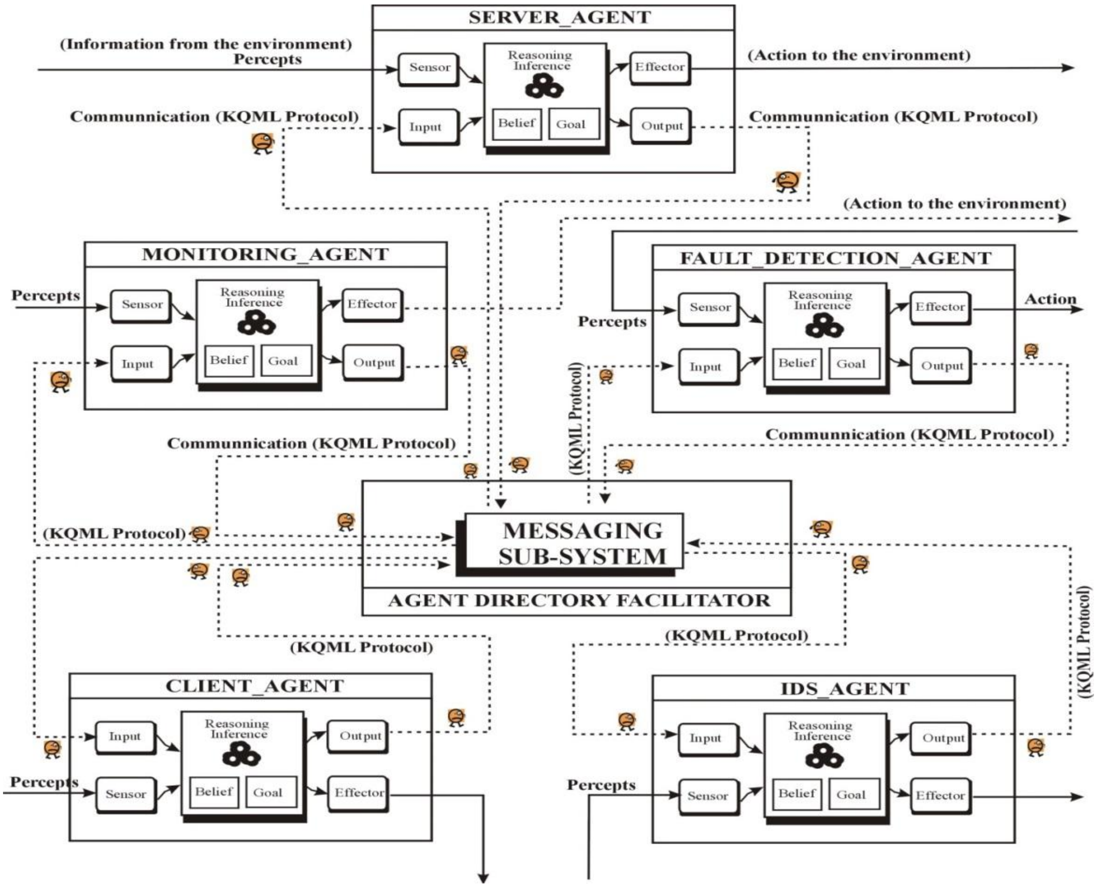

Figure 2 Proposed communication scheme for the multi-agent based system architecture

DOI 10.17148/IJARCCE.2016.57157

Figure 2 shows the proposed communication scheme for the multi-agent based system, in which an agent can communication with other agents in the same platform in order to perform certain task. The communication between agents is made possible by sending messages between them; just as in the case of objects, where messaging is performed by calling objects methods. In the real sense, agents actually do not communicate directly, they rather send a message to a messaging subsystem of their environment and the messaging subsystem delivers the message to the recipient. In this research work, an Agent Directory Facilitator (ADF), with an in-built module for Messaging Subsystem (MS) will be created. The ADF is the main control center for controlling and monitoring the agent's process identity and task configuration, while the (MS) acts as the agents' communication messaging system. For example, if a **Mobile_agent** wants to communicate with the **Server_agent** in order to carry out a specified task; it needs to pass a message first to the Messaging Subsystem (MS) through the agent's output handler, and then the (MS) will send or forward the message to the recipient agent in question through the agent's input handler. In this proposed communication architecture, the communication between agents will be achieved using agent communication languages known as communication protocols. Knowledge Query Manipulation Language (KQML) which defines the character of a message, with its envelope containing how and to whom the message should be delivered to, and the Knowledge Interchange Format (KIF), which handles the message content i.e. passing knowledge which is based on first-order logic are the recommended protocols to aid communication and message interchange in the multi-agent based system design.

2.3 Functions of the various agents

2.3.1Mobile Agent

The mobile agent is the monitoring agent that can be dispatched from the agent management system (server agent) to other nodes in the network environment. It goes into the network to identify with each node in order to collect information about them. At each node, the mobile agent interacts with the client agent. The information gathered at each node with its identities is stored in its database container for onward movement to the next node. This process is repeated until it gets to the last node, at which point the mobile agent migrates back with all the information gathered in its local database container to the server that initially launched it.

The **push migration strategy is adopted in this research work** to design the mobile agent migration system. In the model, when an agent migrates to a new location or node, it carries all its code, data and all state information along. The migration process is divided into three parts.

i. Mobile Agent (mA) starts off from the server (home) platform, S_h and migrates to the first target node in a given hierarchy

ii. Mobile agent migrate from target node N_k to N_{k+1}, where k = 1, 2, …,m-1

iii. Mobile agent migrates back to its home platform.

Accordingly, the total network load of mA is segmented into three parts:

a. The load of mA denoted by B_h while migrating from S_h to N_1

b. The load accumulated by mA is denoted by B_m, while it moves through the target nodes

c. The load of mA denoted by B_f, while it moves from the last node to home node (server)

The total network load denoted by L is therefore given as:

$$L = B_h + B_m + B_f \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \; (1)$$

Let the set of target nodes to be visited be defined as:

$$N = \{N_1, N_2, N_3, \ldots, N_m\} \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \; (2)$$

A mobile agent is composed of the code, data and state information, which are denoted by c, d and s respectively. If the code comprises n-length of code, therefore, the total length of code in bytes is:

$$B_c = c_1 + c_2 + c_3 + c_4 + \ldots + c_{m-1} + c_m \;\ldots\ldots\ldots\ldots......... \; (3)$$

2.3.2 Server Agent

The server agent is a static agent. It executes only on the network administrator's system where the agent management system is installed and resides. It is used to perform the initial configuration processes, network probing, data gathering, analysis and mining functions by the network administrator. It can accept and send requests to and fro other agents. It sends messages to the report handler for onward transmission to the network administrator screen notification system and also integrates with web services for sms alert via mobile phone. The directory facilitator controls and coordinates all the agents operations and processes, while the messaging subsystem assist in agent communications.

2.3.3 Client Agent

The client agent will be installed and will run as a daemon application on all the nodes on the network. It interfaces with each client O. S. in order to fetch and pass instructions to the O.S. It keeps track of data fetched and sent to the O.S. in its local database container for further arguments. It also tries to reason by comparing instructions with what it has in its container in order to make proper decisions, like deciding whether to shut down the node or not. It does interfaces with the mobile agent to transmit or receive messages from the server.

2.3.4 Fault Detection Agent

The fault detection agent will detect faulty links connecting nodes (network cables) and the nodes in the network environment. The detection mechanism contains a pre-classified data from the data patterns concerning various network cables and nodes related problems fully

DOI 10.17148/IJARCCE.2016.57157

classified or segmented into different levels or clusters and it is then stored in its database container. The actual data classification is done using decision tree technique. Agent reasons simply by matching the input stream data with the already classified data. If a match is seen, it flags it as a fault.

2.3.5 Intrusion Detection Agent

The intrusion agent is a hybrid detection mechanism for detecting anomaly node in the network environment. The system is trained to be intelligent for the intrusion detection task using both the supervised and the unsupervised learning techniques in order for the agent to be proactive. For the supervised method, stream of data gathered from the network is classified using decision tree clustering or classification means, and the pre-classified data is stored in a database before onward movement to the engine, while in the case of the unsupervised techniques, the artificial neutral network (ANN) unsupervised kohonen map techniques will be used in the classification of unknown node(s) data frame joining the network, in order to perform profiling for the agent intelligence. After classification by the technique, the agent will then reason by comparing both databases for a match in data patterns in order to achieve its aim of node intrusion detection.

3. RESULTS AND DISCUSSIONS

The following results were obtained and discussed below:

i. The monitoring mobile agent can move from one node to the other as designed in order to shut down nodes on the network using a preset time; the agent can also send a shutdown notification message to notify the network administrator through screen notification system or sms alert via its web services interface.
ii. The detection agents' mechanisms can quickly detect faulty network cable or node as soon as they malfunction or fail and also detect the presence of an anomaly node intrusion in the network environment.

4. CONCLUSION

The multi-agent based system architecture designed for monitoring nodes in a network environment was able to address the network problems identified which is to the network administrator's advantage. The multi-agent based system is recommended for development and practical usage.

REFERENCES

[1] Akinyokun, O. C., Ekuewa, J. B., and Arekete, S. A. (2014) Development of agent-based system for monitoring software resources in a network environment, Artificial Intelligence Research, Published by Sciedu Press, Vol. 3, No. 3

[2] Weiss, (1999) Multi-Agent Systems, MIT Press

[3] Wooldridge, M. (2002) An Introduction to multi-agent systems. John Wiley & Sons Limited England, ISBN: 0-471-49691-X

[4] Russel, S. and Norvig, P. (2003) Artificial Intelligence: A Modern Approach, Englewood Cliffs, New Jersey: Prentice Hall

[5] Macal, C. M. and North, M. J. (2006) Introduction to Agent-based Modeling and Simulation, MCS LANS Informal Seminar slides.

[6] Kok, J. R., and Vlassis, N. (2006) Collaborative Multiagent Reinforcement Learning by Payoff Propagation, Journal of Machine Learning Research 7, pp. 1789-1828

[7] Kakas, A., and Moraitis, P. (2003) Argumentation based decision making for autonomous agents, In AAMAS '03 Proceedings of the second International joint conference on Autonomous agents and multiagents sytems. New York, NY, USA: ACM, pp. 883-890

[8] Jennings, N. R., and Wooldridge, M. J. (1998) Application of Intelligent Agent: Foundations, Applications and Markets, pp. 3-28, Secaucus, NJ, Springer-Verlag, Berlin.

[9] Albert, M., Laengle, T., Woern, H., Capobianco, M. and Brighenti, A. (2003) Multi-agent Systems for Industrial Diagnostics. In Proceedings of 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, pp. 483-488, Washington.

[10] Greenwood, D., Vitaglione, G., Keller, L. and Calisti, M. (2006) Service Level Agreement Management with Adaptive Coordination. Proceedings of the International Conference on Networking and Services (ICNS '06), Silicon Valley, USA.

[11] Hayzelden, A. I., and Bourne, R. A. (2001) Agent Technology for Communication Infrastructures, John Wiley & Sons Ltd, London, UK

[12] Singh, P. K. (2011) Introduction to Computer Networks, V. K. (India Enterprises), Ambala City, India.

[13] Imianvan, A. A. (2009) Development of Mobile Agent for Evaluating the Use of Bandwidth in a Computer Network, PhD Thesis, Department of Computer Science, Federal University of Technology, Akure. In (Akinyokun, O. C., Ekuewa, J. B., and Arekete, S. A., 2014) Development of agent-based system for monitoring software resources in a network environment, Artificial Intelligence Research, Published by Sciedu Press, Vol. 3, No. 3

[14] Arekete, S. A. (2013) Development of a Mobile Agent for monitoring and Evaluation of Activities of Users in a Network Environment, PhD Thesis, Department of Computer Science, Federal University of Technology, Akure, In (Akinyokun, O. C., Ekuewa, J. B., and Arekete, S. A., 2014) Development of agent-based system for monitoring software resources in a network environment, Artificial Intelligence Research, Published by Sciedu Press, Vol. 3, No. 3

[15] Ismail, L., Hagimont, D., and Mossiere, J. (2000) Evaluation of the mobile agent technology: Comparison with the Client/Server paradigm. Information Science and Technology (IST), V ol 19.

[16] Li, C., and Wang, W. (2002) Internet mail transfer and check system based on intelligence mobile agent, Symposium on Applications and the Internet (SAINT) Nara City, Nara, Japan

[17] Dasgupta, D., Gomez, J., Gonzalez, F., Kaniganti, M., Yallapu, K., and Yarramsettii, R. (2003). MDS: Multilevel Monitoring and Detection System, In the Proceedings of the 15th Annual Computer Security Incident Handling Conference, pp. 22-27, The Westin, Ottawa, Ontario, Canada.

DOI 10.17148/IJARCCE.2016.57157