DOI: 10.15625/vap.2020.00150

CLOUD NETWORK MANAGEMENT MODEL BASED ON MOBILE AGENT

Nguyen Minh Phuc[1], Nguyen Ai Viet[1], Tran Quy Nam[2]

[1]Information Technology Institute, Vietnam National University, Hanoi
[2]Posts and Telecommunications Institute of Technology
ngminhphuc@gmail.com , naviet@vnu.edu.vn, namtq@ptit.edu.vn

ABSTRACT: *Individuals and companies are adopting cloud at a faster rate, due to which internet traffic is increasing at a pace which is difficult to manage. With development of new technologies in the cloud we need to alter the traditional protocols to manage the increasing cloud traffic. The paper discusses limitations of one of the main existing network management protocol i.e. Simple Network Management Protocol (SNMP). The proposed Model 'Cloud Network Management Model based on Mobile Agent (CNMMA)' by using Mobile Agent to provide a comprehensive solution model to manage the traffic in cloud and trying to reduce the network traffic to eliminate the packet exchange between manager and agent in SNMP and manage Cloud Network effectively by implement CMIP protocol.*

Keywords: *Cloud Computing, Network Management, Mobile Agent, Virtualization, SNMP, CMIP.*

I. INTRODUCTION

1.1. Data Center Virtualization, Cloud Computing growth and Network Management

The increasing need for data center and cloud resources has led to the development of large-scale public cloud data centers called hyperscale data centers. Networking capabilities are important role for transferring data from and storing data to the cloud. The networking capabilities include networking devices, bandwidth, protocols. Internet carries several types of traffic; the bandwidth of Internet and Cloud Computing are major contributors to this traffic, due to the growing of big data, IoT, social media[1]... The traditional TCP/IP protocols are not being able to cop up with the level of services that cloud requires. For example, when a packet travels on the cloud it may pass through several components before it ever leaves the system: system buffer, transfer application, network stack, VPN, firewalls and filters, network drivers, and the hardware network adapter, Network Devices such as Routers, and Gateways. Each device adds its jitter in processing the packet. Hence, there is very less scope in making delivery fast, so many protocols were created like Multi-Protocol Label Switching (MPLS), Resource Reservation Protocol (RSVP),… that will help in timely delivery of packets with required quality of services [8].

In network management, for better Quality of Services (QOS), system/network administrators should be always aware of the status of the networking devices called agents, including their CPU loads, memory, storage usage etc. Currently, SNMP (Simple Network Management Protocol) has been widely used in remote monitoring of network devices and hosts.

In SNMP, we know that the Network Management Station (NMS) called manager periodically requests or polls the agent. The MIB inside agents contains a counter that counts number of bytes transmitted and received in a time interval on each of its interfaces. The counter is cyclic. The SNMP counters counts only a running total and not the count the number of packets per interval. SNMP manager send polls to agent to compute packets per interval in short duration of time. SNMP polls after every five minutes. Thus, SNMP poller periodically records these counters and collects information[9].

SNMP has advantage that it is an open standard protocol, which is designed to combat the wasted effort and costs when one manufacturer develops its own "proprietary" protocol that only it will support; many devices and manufacturers support it. However, SNMP has many known limitations such as:

- SNMP uses unreliable User Datagram Protocol (UDP) transport, Data may be lost in transit.
- Sometimes an SNMP poller restarts and it loses its track of a counter, counter resets (say after a router reboot), which results in large error in the estimate of traffic. In early versions of SNMP 32-bit counters were used and these counters reset quickly on high speed links. Sometimes SNMP poller wrongly calculates the average rate as per information received, ignoring the missing polls [2].
- "Jitter" caused by polling is another problem in SNMP. The Network Management Station must perform polls to many devices and these polls cannot be performed concurrently. These query –reply packets take some time to transit the network [3]. Finally, the result is that the reply packets reach late due to this jitter. Moreover, routers give low priority to SNMP packets; therefore, they have a delayed response.
- SNMP processes on agents are given low-priority and hence they have a delayed response;
- SNMP is too periodic. Sometimes polling cycles from 30 seconds to several minutes long does not produce the actual picture of the network routing conditions. Even if we speed up the polling cycle it would miss many routing state changes and would generate much management traffic overhead [4].

- SNMP communication delays the action to be taken by manager, as manager has to first send a query message in which it has to access the MIB , the object data then travel all the way to manager and if required send the update message to manager. Thus, using SNMP is not meant for very large networks because sending a packet to get another packet causes delay in communication and hence in management. This type of polling causes large volumes of regular messages and end in problem response times that may be unacceptable [9].
- There is no acknowledgement for Trap messages in SNMP. If UDP is used with Internet Protocol (IP) to deliver trap message by agent, the agent gets no response whether the trap message has been delivered to manager or not. This is unacceptable for such critical messages.
- SNMP does not directly support crucial commands. The only way to prompt an event at an agent is indirectly by setting an object value. A more efficient way is to use remote procedure calls with parameters, conditions, status and results, that SNMP does not support.
- SNMP marginal errors should not be ignored as feeding such small errors into management process causes major problems, corrupting the results and leads to poor management.
- SNMP does not fully support all functions of network management such as CMIP protocol such as: CM-Get, M-Set, M-Action, M-Create, M-Delete, M-Event-Report.

II. CLOUD NETWORK MANAGEMENT MODEL BASED ON MOBILE AGENT (CNMMA)

In previous session, we studied the weakness of SNMP in network management communication, so we we will introduce a new network management model that will try to overcome those problems. Network Management usually requires the support of an agent in the managed host, and the database in the agent provides the management information needed for a management application, but in our model based on Mobile Agent, network management don't have to use agent.

2.1. The CNMMA Model Architecture

The cloud network management CNMMA Model is based on Mobile Agent technology, network manager station, virtualized server manager relationship and combination of network management protocol (CMIP protocol) and ACL specification (Mobile Accumulative language by FIPA Specification for Mobile Agent). The components of CNMMA Model included basic core components:

- Mobile Agent Platform
- Virtualized Network Management Server
- The Network Management Station (NMS)
- Network Management Mobile Agent (NMMA)

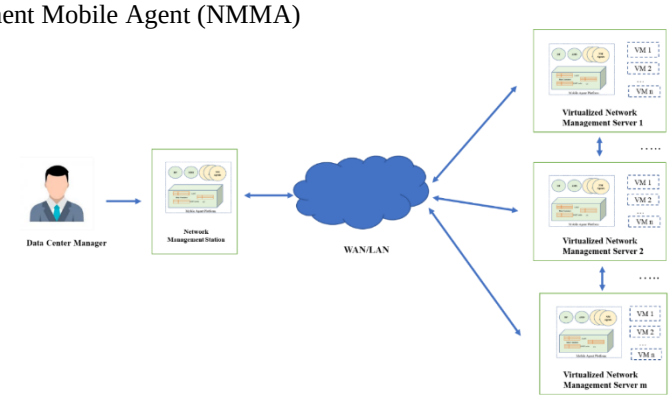

Figure 1. The CNMMA Model Architecture

2.2. Components of CNMMA Model

1. Mobile Agent Platform: is a platform which provides Mobile Agent-based application, an infrastructure for building and executing mobile agents for network processing. Mobile Agent Platform has three core functions includes:

– *Agent Hierarchy Management:* Each Mobile Agent system corresponds to the root node of an agent hierarchy, which is maintained as a tree structure in which each node contains a mobile agent and its attributes. Agent migration in an agent hierarchy is performed simply as a transformation of the tree structure of the hierarchy [7].

– *Agent Execution Management*: Each agent can have more than one active thread under the control of the system. The system maintains the life-cycle state of agents. When the life-cycle state of an agent is changed, for example, at creation, termination, or migration, the system issues certain events to invoke certain methods in the agent and its containing agents [7].

– *Agent Serialization and Security Management*: The system has a function for marshaling agents into bit streams and unmarshaling them later. The system verifies whether a marshaled agent is valid or not to protect the system against invalid or malicious agents, by means of security mechanism [7].

2. Virtualized Network Management Server (Manager): is used to manage and supervise the entire network. It receives all the information and displays it. It may be a pool of virtualized servers. We can take cloud services for Obtaining Manager Services. We assume that Manager is virtualized pool of servers kept on cloud. The Manager is also installed Mobile Agent Platform to make a Mobile Agent Platform Networking [2].

3. The Network Management Station (NMS): A network node that contains a CNMMA Mobile Agent Platform to create environment enable Network Management Mobile Agent can run and managed in its.

4. Network Management Mobile Agent: is Mobile Agent which written to operation such as create, migration, delete and do network management functions in Mobile Agent Platform.

2.3. The CNMMA Protocol

2.3.1. Compare SNMP Protocol vs CMIP Protocol for network management protocol in CNMMA model

CMIP and SNMP, each of them has some advantages and weaknesses. Selection one of these protocols depends on the many parameters. Simplicity in the design is the main advantage of SNMP. Also, when the communicated messages between manager and agent entities are low, SNMP will utilize. Low security is the main weakness of SNMP that recently is going to be better in the new versions. Because of using connectionless services in the transport layer, agent entity will not receive acknowledge, therefore it cannot sure that its alarm report reached to manager entity. Therefore, when there is too much management information to communicate, the network traffic load rises and might be management information lost.

On the other hand, to compensate lack of SNMP, CMIP has designed and it can be used for lager networks. Object-oriented model will use to design and implement SNMP [2]. The main advantage of CMIP is its ability to define the techniques to cover manual control, security and filtering of the management information. The main weakness of CMIP is resource occupation time which is more than SNMP [13]. Table 1 summarize the list of the CMIP and SNMP services.

Table 1. CMIP and SNMP services

Name	Valid Service(s)	Description
Get	CMIP and SNMP	Obtain a value maintained by the managed object.
Set	CMIP and SNMP	Set a value maintained by the managed object.
Event Report	CMIP and SNMP	Report special conditions about a managed object.
		Name and value of the next SNMP attribute will determine in the object.
		One of the actions defined for the managed object will invoke.
Create	CMIP	Instance of an object class will create.
Delete	CMIP	Delete an instance of class.

Hence, we choose CMIP protocol to working with CNMMA model because its overcome SNMP limitation and also avoid transferring large network management object frequently while using CMIP protocol by using Mobile Agent characteristic as its advantages.

2.3.2. ASN.1/GDMO to define Network Management Object

Abstract Syntax Notation One (ASN.1) is a standard interface description language for defining data structures that can be serialized and deserialized in a cross-platform way. It is broadly used in telecommunications and computer networking, and especially in cryptography. Because ASN.1 is both human-readable and machine-readable, an ASN.1 compiler can compile modules into libraries of code, CODECs, that decode or encode the data structures. Some ASN.1 compiler can produce code to encode or decode several encodings, e.g. packed, BER or XML. The latest revision ASN.1 standard is X.680 series of recommendations is the 5.0 Edition, published in 2015. CMIP models management information in terms of managed objects and allows both modification and performing actions on managed objects. Managed objects are described using GDMO (Guidelines for the Definition of Managed Objects) and can be identified by a distinguished name (DN), from the X.500 directory.

The Guidelines for the Definition of Managed Objects (GDMO) is a specification for defining managed objects of interest to the Network Management for use in CMIP. GDMO is similar to the Structure of Management Information (SMI) for defining a management information base (MIB) for SNMP. For example, both represent a hierarchy of managed objects and use ASN.1 for syntax. GDMO is defined in ISO/IEC 10165 and ITU-T X.722.

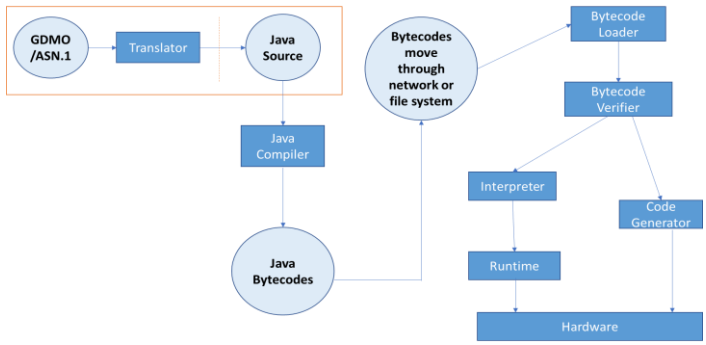

Figure 2. Flows of compiled ASN.1/GDMO to network management object

2.3.3. ACL (Mobile Agent language FIPA Specification)

Each Mobile Agent can travel though Mobile Agent Platform, each Mobile Agent have to written in a compliance language which is constructed by FIPA (Foundation for Intelligent Physical Agents) Standard. A FIPA ACL message contains a set of one or more message parameters. Precisely which parameters are needed for effective agent communication will vary according to the situation; the only parameter that is mandatory in all ACL messages is the *performative*, although it is expected that most ACL messages will also contain *sender*, *receiver* and *content* parameters [12].

2.4. Network Management operations

For better QoS of cloud services, system/network administrators of network should also be always aware of the current status of the manager machines in the cluster, including their CPU loads, storage usage, and network utilization. Furthermore, administrators are also interested in how many Virtual Machine (VM) instances are allocated in a manager host machine and how well each VM instance is running. As a great number of managers that are deployed to provide virtual machines to a variety of agents.

The definition of the operation and the consequent behavior of the communicating entities is dependent upon the specification of the managed object at which the operation is directed and is outside the scope of the Common Management Information Services (CMIS). However, certain operations are used frequently within the scope of systems management and CMIS provides the following definitions of the common services that may be used to convey management information applicable to the operations:

- The M-GET service is invoked by a CMISE-service-user to request the retrieval of management information from a peer CMISE-service-user. The service may only be requested in a confirmed mode, and a reply is expected [11].
- The M-SET service is invoked by a CMISE-service-user to request the modification of management information by a peer CMISE-service-user. The service may be requested in a confirmed or a non-confirmed mode. In the confirmed mode, a reply is expected [11].
- The M-ACTION service is invoked by a CMISE-service-user to request a peer CMISE-service-user to perform an action. The service may be requested in a confirmed or a non-confirmed mode. In the confirmed mode, a reply is expected [11].
- The M-CREATE service is invoked by a CMISE-service-user to request a peer CMISE-service-user to create an instance of a managed object. The service may only be requested in the confirmed mode, and a reply is expected [11].
- The M-DELETE service is invoked by a CMISE-service-user to request a peer CMISE-service-user to delete an instance of a managed object. The service may only be requested in the confirmed mode, and a reply is expected [11].
- The M-CANCEL-GET service is invoked by a CMISE-service-user to request a peer CMISE-service-user to cancel a previously requested and currently outstanding invocation of the M-GET service. The service may only be requested in the confirmed mode and a reply is expected [11].

2.5. CNMMA Model Security

The dynamic nature of mobile agent led to complex design and security threats. These threats are categorized in four categories according to the originator of the threat and the victim. These categories are listed below:

1. **Agent to Platform:** This category is related to threats of a mobile agent to a particular platform.
2. **Platform to Agent**: This category is related to threats of platform to a particular mobile agent.
3. **Agent to Agent:** This category is related to threats that may occur due to the interaction among mobile agents.
4. **Platform to Platform:** This category addresses threats among platforms of mobile agents. [9]

Table 2. Summary of threat classes and corresponding attacks

	Threat class			
Attack	**1**	**2**	**3**	**4**
Masquerading	***Yes***	***Yes***	***Yes***	***Yes***
Unauthorized	***Yes***	***Yes***	***Yes***	***No***
Denial of Service	***Yes***	***Yes***	***No***	***No***
Repudiation	***Yes***	***Yes***	***No***	***No***
Alternation	***No***	***Yes***	***Yes***	***No***
Eavesdropping	***No***	***Yes***	***No***	***No***

The mobile Agent security countermeasures:

We provide the basic requirements for securing mobile agent systems, these includes: authentication, confidentiality, availability, and accountability and non-repudiation.

- Authentication and Authorization: Authentication is the process of verifying the identity of the entity. In mobile agent systems, authentication process requires both agent and platform to be authenticated by each other. i.e. the agent knows the executing environment and the executing environment knows the agent. Authorization is the process of deciding to grant a request or not after entity has been authenticated. To achieve those security properties, digital signatures and password protection are used together [9].
- Confidentiality, Privacy, and Anonymity: Confidentiality refers to the state of hiding sensitive data from being disclosed to un-authorized parties. The disclosure of such data may degrade the privacy level since data may have private information about agent. Revealing the behavior of a mobile agent may also degrade privacy to some extent. The privacy concerns may be treated by means of privacy preserving techniques such as enforcing an anonymity level which might reduce the threat. Encryption also works well for hiding sensitive data from un-authorized parties but it may degrade performance [9].
- Accountability and Non-repudiation: The problem of repudiation arise when a party claims being not involved in activity or a communication while it actually did. To overcome this problem, important communications and security related activities should be securely recorded for auditing and tracing purposes as well as for non-repudiation. These logs must be protected from un-authorized access to maintain the privacy and security levels of the system[9].
- Availability: A mobile agent platform should ensure the availability of data and services required for local and incoming mobile agents. This implies that the platform should provide controlled concurrency, simultaneous access, deadlock management and exclusive access when required [9]. Platforms should also be able to detect and recover from software crashes as well as hardware failures. It should also deal well and defend against denial of Service attacks.
- Security Threat Detection and Prevention Mechanisms in Mobile Agent Systems: In this sub-section, we will list some proposed mechanisms to detect or prevent an attack on mobile agent systems. These mechanisms are used to ensure that platforms will respect the policies of mobile agents they serve [9].
- Detection Techniques: Detection techniques are used to find out whether an agent has been changed or not. This includes tampering code, state, or execution flow. These techniques varies according to whether they work automatically or not, whether they work during execution or after termination, and whether they detect all possible alternations or some of them [9].
- Detection mechanisms also varies according to the scope of detection, some techniques use range checkers which detects illicit code manipulation according to variable values or timing constraints. Others use execution tracing and cryptography that allows them to detect attacks against code, state, and execution flow of mobile software components. Hash function has been proposed as a detection mechanism for protecting the forward integrity results gathered by mobile programs [9].
- Prevention Techniques: These mechanisms aim to leverage mobile agent security level against tampering attacks. Using these techniques will make it very difficult to illegally access or modify the code. Prevention techniques varies according to the goal of prevention which includes: preventing the entire agent or part of it, preventing attacks permanently or temporary, and trusting some functionalities or no trust is assumed [9].

2.6. Benefits of the Proposed Model

- **Reduced communication costs:** The latency and network traffic of interactions often seriously affect the quality and coordination of two programs running on different computers. As in Mobile Agent platform, if one of the programs is a mobile agent, it can migrate to the computer the other is running on communicate with it locally. That is, mobile agent technology enables remote communications to operate as local communications.
- **Asynchronous execution**: After migrating to the destination-side computer, a mobile agent does not have to interact with its source-side computer. Therefore, even when the source can be shut down or the network between the destination and source can be disconnected, the agent can continue processing at the destination. This is useful within unstable communications, including wireless communication, in smart environments [5].

- **Direct manipulation:** A mobile agent is locally executed on the computer it is visiting. It can directly access and control the equipment for the computer as long as the computer allows it to do so. This is helpful in network management, in particular in detecting and removing device failures. Installing a mobile agent close to a real-time system may prevent delays caused by network congestion [5].
- **Dynamic-deployment**: It's as useful as a mechanism for the deployment of software, because they can decide their destinations and their code and data can be dynamically deployed there, only while they are needed [5].
- **Easy-development of distributed applications:** Most distributed applications consist of at least two programs, i.e., a client-side program and a server-side program and often spare codes for communications, including exceptional handling. However, since a Mobile Agent itself can carry its information to another computer, we can only write a single program to define distributed computing. [5]
- **Full of control network management**: Implement full operations of CMIP protocol for better commands on agents than SNMP protocol such as M-Get, M-Set, M-Action, M-Create, M-Delete, M-Event-Report.

III. IMPLEMENTATION

To demonstrate our model is working effectively, we do lab experiment on practical model by using open source Mobile Agent Platform to create an network management application, implement protocol and network infrastructure to simulate.

3.1. JADE – A Mobile Agent Platform

JADE is a Mobile Agent platform framework, which is a composed of agent containers that can be distributed over the network. Agents live in containers which are the Java process that provides the JADE run-time and all the services needed for hosting and executing agents. There is a special container, called the main container, which represents the bootstrap point of a platform: it is the first container to be launched and all other containers must join to a main container by registering with it. The UML diagram in Figure 3 schematizes the relationships between the main architectural elements of JADE.

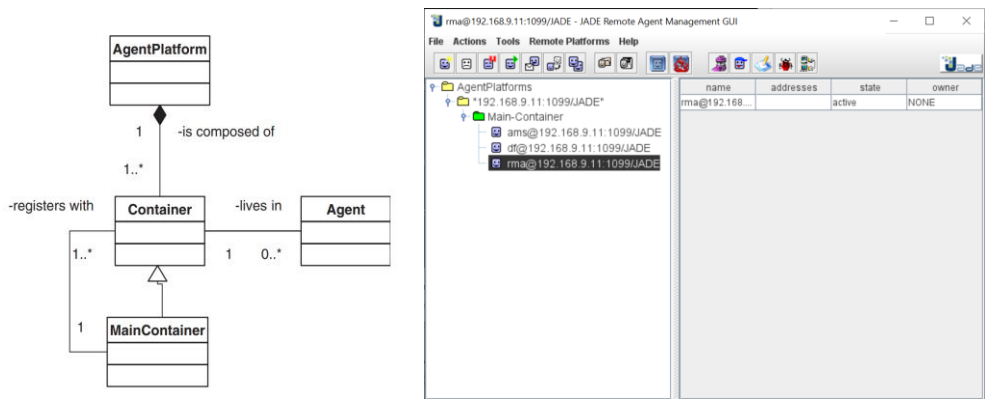

Figure 3. Relationship between the main JADE architectural elements & JADE Framework GUI

As an implementation the CNMMA model, we proposed the runtime architecture base on the JADE platform as Figure 4 CNMMA model implementation based on JADE platform framework below.

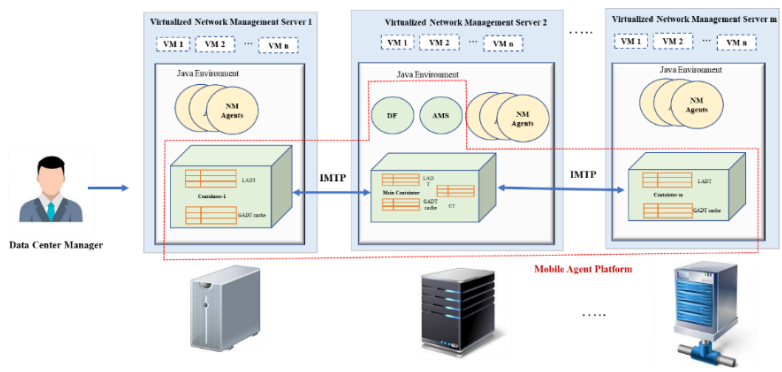

Figure 4. CNMMA model implementation based on JADE platform framework

3.2. ASN.1 for CMIP protocol

We also studied ASN.1 abstract syntax definition for CMIP protocol and we create basic functions of CMIP protocol such as:M-Get, M-Set, M-Action, M-Create, M-Delete, M-Event-Report based on ITU-T X.10 (ISO/IEC 9595 :1998).

Table 3. Example of ASN.1 code for CMIP protocol

```
-- CMISE operations
-- Action operation (M-ACTION)
m-Action OPERATION ::= {
   ARGUMENT              ActionArgument
   RETURN RESULT       FALSE
   ALWAYS RESPONDS    FALSE
   CODE                    local:6
}
```

```
m-Action-Confirmed OPERATION ::= {
   ARGUMENT    ActionArgument
   RESULT       ActionResult
   OPTIONAL    TRUE -- this result is conditional;
   -- for conditions see 8.3.3.2.9 of ITU-T Rec. X.710
   ERRORS
      {accessDenied | classInstanceConflict | complexityLimitation |
invalidScope
         | invalidArgumentValue | invalidFilter | noSuchAction |
noSuchArgument |
         noSuchObjectClass | noSuchObjectInstance |
processingFailure |
         syncNotSupported}
   LINKED       {m-Linked-Reply}
   CODE          local:7
}
```

3.3. ACL In JADE platform

Agent communication is probably the most fundamental feature of JADE and is implemented in accordance with the FIPA specifications. The communication paradigm is based on asynchronous message passing. The particular format of messages in JADE is compliant with that defined by the FIPA-ACL message structure [6]. Each message includes the following fields[6]:

- The sender of the message.
- The list of receivers.
- The communicative act (also called the 'performative') indicating what the sender intends to achieve by sending the message.
- The content containing the actual information to be exchanged by the message.
- The content language indicating the syntax used to express the content. Both the sender and the receiver must be able to encode and parse expressions compliant with this syntax for the communication to be effective.

3.4. Network management simulator

For experiment the CMIP protocol prototype and theory of CNMMA Model, we created a network simulator based on GNS3 and make a connection from simulate networked to a physical Virtualized PC which installed JADE platform framework for testing create and migration Network Management Mobile Agent.

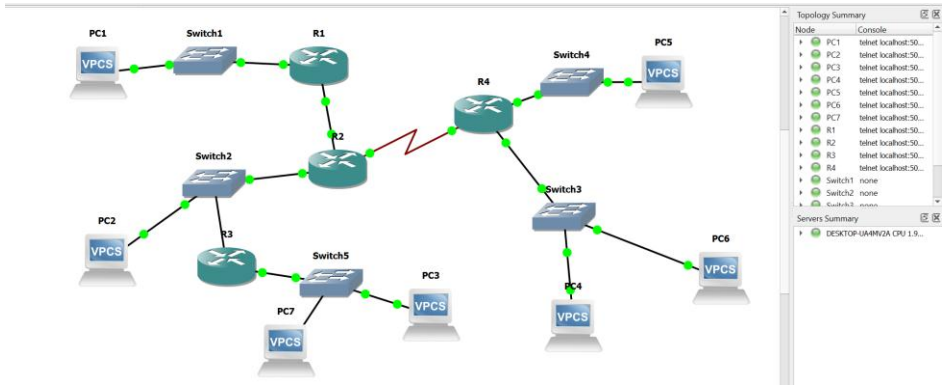

Figure 5. Network management simulator using GNS3

IV. CONCLUSIONS

Cloud Network Management Model is such CNMMA Model that efficiently manages cloud traffic with more accurate results and providing security to the network management. Though, the paper describes the Cloud Network Management Model at abstract level, lab experiment and not fully implementation. Our future research work will concentrate on describing each part of the Model at more detailed design and make a full functionality of the model. In continuation we will focus on analyzing the use of SDN to check the flows of traffic in cloud and try to do pilot test in production environment.

V. REFERENCES

[1] Cisco White Paper, (November, 2016), "Cisco Global Cloud Index, 2015-2020".

[2] Ya-shiang peng, Yen-cheng Chen, "SNMP-based monitoring of Heterogeneous virtual infrastructure in Clouds", 2011.

[3] J. Swarna, C. Senthil raja, Dr.K.S.ravichandran, "Cloud monitoring based on SNMP", 2012.

[4] Laurent Andrey, Olivier Festor, Abdelkader Lahmadi, Aiko pras and Jürgen Schönwälder, "Survey of SNMP performance analysis studies", International Journal of Network Management, 19 pp. 527-54, 2009.

[5] Ichiro Satoh, Mobile Agents, August 2010.

[6] Fabio Bellifemine, Giovanni Caire, Dominic Greenwood, John Wiley & Sons Ltd, Developing Multi-Agent Systems with JADE, 2007.

[7] Ichiro Satoh, "Building Reusable Mobile Agents for Network Management", IEEE Transactions on Systems, Man and Cybernetics, Vol.33, No. 3, part-C, pp. 350-357, August 2003.

[8] Dr. Mamta Madan, Mohit Mathur, "Cloud network management model – a novel approach to *manage cloud traffic", International Journal on Cloud Computing: Services and Architecture (IJCCSA),Vol. 4, No. 5,* October 2014.

[9] Belal Amro, (October 2014), Mobile Agent Systems, Recent Security Threats and Counter Measures.

[10] Ichiro Satoh, Springer, "Building and Selecting Mobile Agents for Network Management", Journal of Network and Systems Management, Vol.14, No.1, pp. 147-169, 2006.

[11] CCITT, X.710 and ISO/IEC 9595: Information technology - Open Systems Interconnection - Common management information service definition, 1991.

[12] FIPA, FIPA ACL Message Structure Specification - SC00061G, December 2002.

[13] Gholamreza Farahani, Rahani, Gholamreza, "New proposed architecture for Q3 interface to manage IP-based networks", International Journal of Computer Networks & Communications (IJCNC), Vol.9, No.4, July 2017.

MÔ HÌNH QUẢN LÝ MẠNG CHO ĐIỆN TOÁN ĐÁM MÂY DỰA TRÊN NỀN TẢNG TÁC TỬ DI ĐỘNG

Nguyễn Minh Phúc, Nguyễn Ái Việt, Trần Quý Nam

TÓM TẮT: *Các doanh nghiệp, tổ chức và các cá thể đã dịch chuyển sang môi trường Điện toán đám mây (Cloud Computing) với tốc độ cao, điều này làm gia tăng băng thông mạng và Internet rất lớn và dẫn tới sự khó quản lý về mạng cũng như băng thông mạng. Cùng với sự phát triển của điện toán đám mây, chúng tôi nhận thấy cần phải thay đổi cách thức quản lý mạng truyền thống hiện nay để quản lý mạng cho Điện toán đám mây hiệu quả hơn. Trong bài báo này, chúng tôi đưa ra các giới hạn của các giao thức mạng truyền thống hiện nay chẳng hạn như Giao thức quản lý mạng đơn giản (SNMP), và đề xuất mô hình quản lý mạng mới cho Điện toán đám mây "Mô hình quản lý mạng cho Điện toán đám mây dựa trên nền tảng Tác tử di động" thông qua nền tảng Tác tử di động (Mobile Agent). Mô hình này cung cấp một giải pháp toàn diện để quản lý mạng và giao thông mạng cho Điện toán đám mây, giảm thiểu nghẽn giao thông mạng bằng cách loại bỏ các gói tin trong quá trình trao đổi dữ liệu giữa trình quản lý (manager) và các tác tử (agent) trong giao thức SNMP và quản lý hệ thống mạng cho Điện toán đám mây hiệu quả hơn qua việc cài đặt giao thức quản lý mạng CMIP.*