ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

A Study of Distributed Network Management Architectures

Kriti Nagpal[1] & Anish Saini[2]

MTech Student, Department of Computer Science & Engineering, Echelon Institute of Technology, Faridabad, India[1]

Assistant Professor, Department of Computer Science & Engineering, Echelon Institute of Technology, Faridabad, India[2]

ABSTRACT: Initially network was managed by centralized client server technology using various protocols like SNMP, CMIP etc. But on large scale, it has many disadvantages like problem of traffic, failure of server etc as centralized network is not scalable and flexible enough. So we need a better framework to improve the management services of a network in a more precise way. This leads to the management of network in distributed way. There are already various architectures designed for distributed network management. Some of them are Client Server mobile agent framework, MAGENTA, NetPatrol etc. This paper discusses the various architectures of distributed network system and understands the working of each model. Also discuss the advantages and disadvantages of one model over other.

KEYWORDS: Network Management; SNMP; Mobile Agents; Architecture

I. INTRODUCTION

Network management is a way to configure, maintain, operate and administrate the network system. A way of characterizing network management is FCAPS - Fault, Configuration, Accounting, Performance and Security [10]. Functions of network management also include controlling and monitoring network resources. Typically network management is based on the centralized client-server architecture having simple network management protocol (SNMP). Here manager application is the client and static mobile agents are distributed servers. Physical resources known as objects are grouped into tree structured management information base (MIB). Problem with SNMP is that it is not scalable when the size and complexity of network increases which will increase gradually. Thus there is need to decentralized distributed network management.

Another type of management protocol is common management information protocol (CMIP). CMIP models management information in terms of managed objects and allows both modification and performing actions on managed objects. CMIP was designed to challenge the impact of SNMP. So to achieve this it has more inbuilt features than SNMP. For example, SNMP defines only "set" actions to alter the state of the managed device, while CMIP allows the definition of any type of action. CMIP was a key part of the Telecommunications Management Network, and enabled cross-organizational as well as cross-vendor network management. However, most TCP/IP devices support SNMP and not CMIP. This is because of the complexity and resource requirements of CMIP agents and management systems. CMIP is supported mainly by telecommunication devices.

The mobile agent (MA) technology has been proposed for managing the networks and distributed system to overcome the scalability problem of centralized paradigm.

MA refers to a software unit travelling between network nodes carrying its persistent state and code. MA can follow either pre defined path or context-dependent itinerary. There are two types of scenarios involve in MA technology, (a) multi-hop MAs that sequentially visit a set of nodes to retrieve and filter management data and then deliver high-level results to the manager host, or (b) single-hop agents uploaded to remote de-vices where they monitor systems and notify the manager in the event of a 'system's health parameter' [6] crossing a pre-determined threshold.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

In this paper, the different types of architecture for distributed network management have been discussed.

II. RELATED WORK

A typical network management system comprises of one or more than one network management stations (NMS), MIB, GUI and Database utilizing a particular protocol like SNMP, CMIP, SNA etc. SNMP [17] is considered as network management protocol for explaining the details of network management functionalities.

In peer network management there is set of NMSs which manage the different domains of the network with periodic interaction among them [3]. Here DB is partitioned and replicated between peer NMSs. SNMP agents are used for communication between peer NMSs.

The advent of MA has revolutionized the network management protocols. Because of MA technology, management of distributed network has become easy and flexible. Also implementing MA on the network is very simple. Ma carries state and code to different hosts and according to situation they can adapt. According to use there are two types of platform to implement MA known as mobile agent platform (MAP).

(i) General purpose MAP e.g. IBM Aglets , Toshiba Bee- gent , Present Technologies JAMES. Although general purpose MAPs are easy to implement but they share certain weaknesses i.e. missing functionalities, unnecessary features, large MA size etc. Thus to eradicate these problems, second type of MAP came into existence.

(ii) customized/optimized MAP e.g. CodeShell [5], MobileSpaces [16].

However these customized MAPs also have some drawbacks.

(i) Heavy weight as MA is carrying codes [13][16].
(ii) Takes efforts for management tasks [5].
(iii) Do not address security issues.

Further MA used in mobile management network is implemented by MAGENTA (Mobile Agent environment for administration) [15]. MAGENTA provides extensibility and modifiable functionalities of the agents [2]. The protocols used in managing the mobile network is SNMP [17] and the OSI management system which utilizes the CMIP [17]. James White [9] introduces MA as a strategy for distributed applications.

Other applications of mobile agent will be in Service Oriented Architecture (SOA). SOA uses Web Services (WS) which are software applications accessible through a URL. WS with the help of XML based protocols like SOAP get executed when WS client request WS. Interaction between WS and WS client is through interface.

III. CLIENT SERVER ARCHITECTURE

Client server architecture (CSA) in distributed environment is the core computing concept. CSA use many technologies and protocols like HTTP, DNS, SMTP, Telnet etc. Client may include web browsers, email software, chat applications etc and server may include web, database and email among others.

There are multiple ways to implement CSA. We are discussing about two main architectures (i) Centralized Network Management System and (ii) Peer Network Management System.

A. *Centralized Network Management System:*

In CSA, server manages processes and stores all data, and client requests specified processes and data. Client sometimes handle processing but it needs server data resources for the completion of process. This CSA is basically centralized network shown in figure 1[5]. Here there is only one centralized server that provides data to the clients. The exchange of data takes place with the help of mobile agent in both the direction i.e. server to clients and vice versa.

Problem with CSA is that when there are number of clients requesting, the server shut down or not function properly. As CSA is centralized network management technique, shutting off servers led to the failing of this scheme. Thus the primary need to improve the architecture was to decentralize the architecture.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

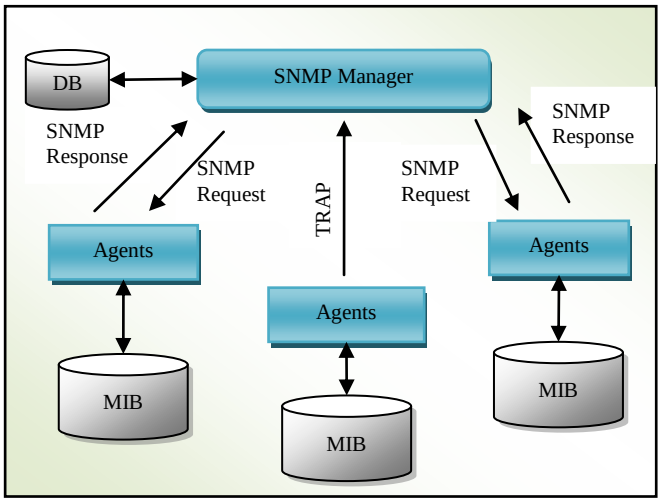

Fig 1. Centralized NMS architecture

B. *Peer Network Management System:*

This is very simple kind of network. Consider a scenario where you can connect two computers that can share resources. These computers do not require connecting to server computer, thus this form a peer network. Peer network can be an ad hoc connection by connecting couple of computers via USB or it can be a permanent connection just like computers in workstations.Access rights are governed by setting sharing permissions on individual computers. For example, if User A's PC is connected to a printer that User B wants to access, User A must set his machine to allow (share) access to the printer. Similarly, if User B wants to have access to a folder or file, or even a complete hard drive, on User A's PC, User A must enable file sharing on his PC. Access to folders and printers in an office can be further controlled by assigning passwords to those resources.

In peer network shown in figure 2[2], each workstation has equal responsibilities and same capability. It is often compared and contrasted to the classic client/server architecture, in which some computers are dedicated to serving others.Peer to peer infrastructure provides multipoint communication as well. It allows the nodes to pass data to and from other nodes in the form of a record. In peer to peer network each computer connected is at once a client and server both. No special network operating system is required in this infrastructure.

The major disadvantage of this architecture is that it is very difficult to administrate. One person cannot determine the whole accessibility setting of whole network. Also there was no way to backup or recover a lost file. Transferring viruses and other malware in this network is very easy.

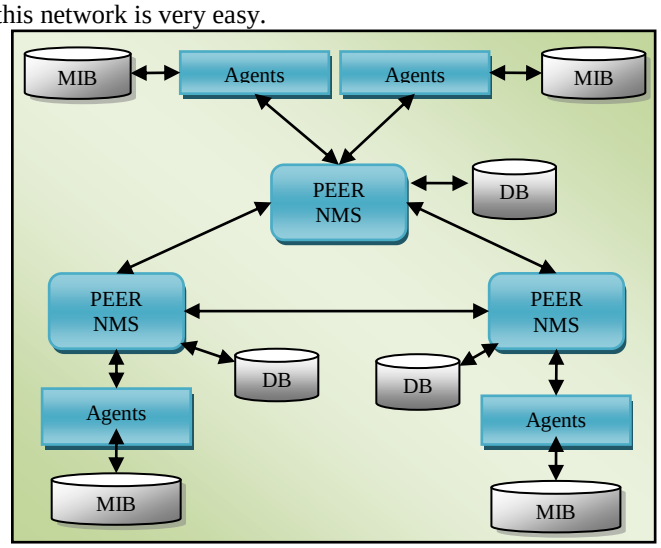

Fig 2. Peer NMS architecture

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

IV. CLIENT SERVER ARCHITECTURE WITH MOBILE AGENT

The main design objective of using mobile agent is code reusability and addition of new services easily. Mobile agents provide the advantages of Distribution of management code, Decentralization, Monitoring and statistics for large SNMP variables, High speed networks.

The modular design of CSA with MA consist of four components- (a) the Manager, responsible for launching and controlling MAs, (b) the MAs to migrate between managed entities, (c) the mobile agent server (MAS), (d) the mobile agent generator(MAG).

The network administrator creates MA using MAG tool. This MA is first stored in MA code repository. When the monitoring task will start then this MA will be retrieved by the manager application. This MA will be carrying an itinerary that is predefined route and starts visiting the group of nodes sequentially. Then this MA will be received by local MAS. The MA will retrieve management data and then hop on to next node. Ultimately it will return back to the manager station with collected data. MA should be able to collect data from different nodes irrespective of different hardware platform or operating system. Thus MAP has been in java. Figure 3[5] shows the typical scenario of MA based architecture.

Manager Application

The main function of manager application is to monitor and control operations on running processes. The components of manager applications are

Network discovery thread (NDT)

At first NDT will execute topology discovery algorithm and will create a list of discovered active MASs. Active servers can be discovered either on manager's initialization or when new MAS start operation. Manager application is notified and host name is included in the list, in the latter case.

Itinerary Scheduler module (ISM)

This module decides number of MAs that will be needed for a process and also decides the route to be followed by MAs. The ISM follows heuristic algorithm for itinerary planning, which has been incorporated into the manager's ISM component.

Polling Threads

Each of the polling thread is correspondent to the individual monitoring task. It also initialized MA and provide them itinerary. Whenever manager application shuts down, these threads are stored as polling thread configuration object and as soon as the application starts again, these threads will resume their work.

Problem with this architecture was that this method is not secure enough; also mobile agent carrying data on different nodes will cause higher weight which can be inflexible.

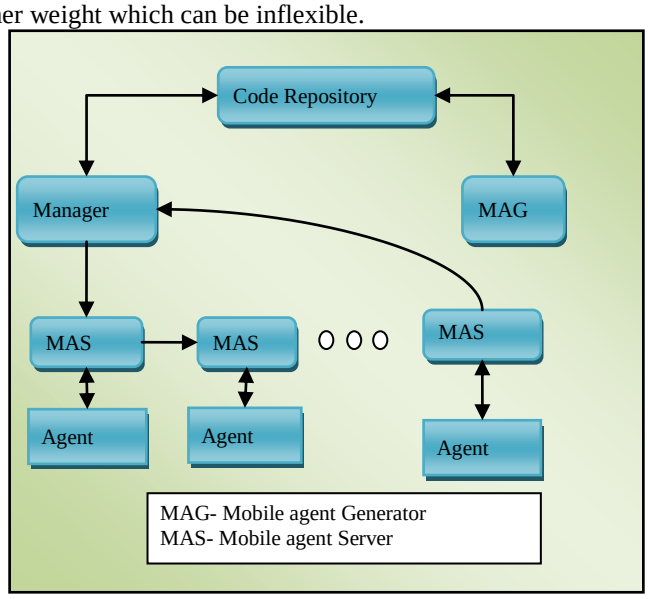

Fig 3. The mobile agent infrastructure

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

V. MAGENTA

MAGENTA is mobile agent environment for distributed administration. It is implemented on JAVA as it is platform independent language. MAGENTA COMPRISES of lieu and agents. Lieu are small programs that works as supporting services for receiving, executing, storing and moving the incoming messages. It also provides communication between agents. All lieu's have unique name & they can appear or disappear dynamically. MA in MAGENTA works similarly as other MA but the difference is that here MAs are adaptive in nature. That means they follow the itinerary but according to the environment, they will change their path, to communicate and exchange notes. The agent first detects the changes and then save it to variable knowledge. MAGENTA also provides features to support mobile computing.

MAGENTA provides agents for

(i) Remote Execution

(ii) Remote Evaluation

(iii) Code on demand

Remote execution is used to initialize an agent and send it to remote site.

Remote evaluation is resourceful for constrained devices which do not support Java environment or can execute lieu.

Code on demand is a variant of agent mobility paradigm where a site requests another site to send an agent with the desired code.

MAGENTA also provides a way to lieus to dynamically appear or disappear. This can be preplanned or unplanned. Preplanned mechanism provide exit to the lieu's. Then this lieu broadcast this message to other lieus and will be disappeared. When the new lieu appear then the update of this information is carried out to notify others.There is also a directory present in MAGENTA services which keeps the records of agents. Every lieu can requests other lieu about the agents launch by them. Also MAGENTA provides a flexible architecture as code can be executed by agents by carrying it or it can be executed directly from remote site. Thus according to the resources available code can be executed.Naming scheme is being used in MAGENTA to provide unique name to lieu. It can be done either automatically or at the time of commencement. The name will remain unchanged till lieu gets destroyed. Even on the change of IP address, lieu name will remain same. This scheme helps agents to identify the correct lieu.

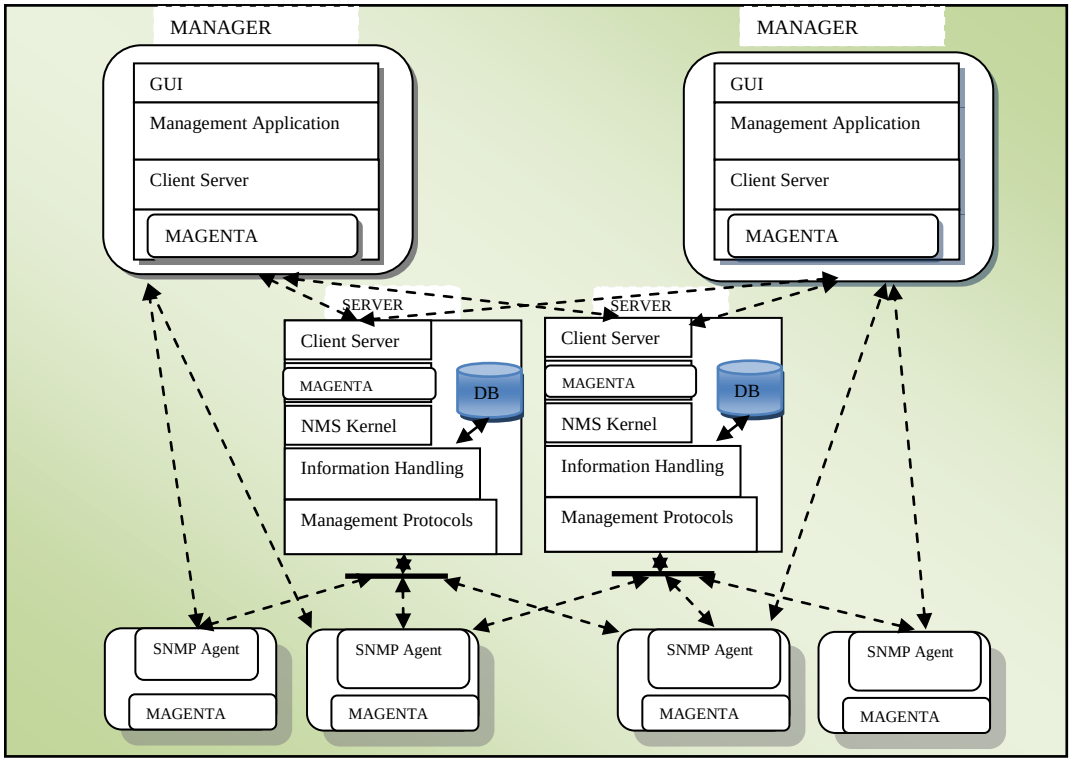

Fig 4. MAGENTA

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

At the time of system crash or any other problem, lieus get disappear which causes loss of data. Thus to avoid this, backup copies are maintained in MAGENTA. This is done by creating a backup copy by lieu, before it sends data to next agent. When agent receives the file, the next lieu send message to previous lieu to destroy the backup data in order to avoid number of copies. This scheme is known as tolerating faults in MAGENTA.MAGENTA shown in figure 4[1][3] provides multiple services like traceability of agents, flexibility of location of code and system independence. A directory is maintained in MAGENTA to trace the agents. This helps the lieu to determine the status of the agent and send the information accordingly. Another important feature is that either agent can execute the code from the remote server or it can carry the code within. According to the availability of resources agents can execute the code. In order to implement MAGENTA independent of platform, it is written in Java. This helps to maintain system independence as Java is platform independent language.

VI. NETPATROL

The convergence of web servers with MAs and SNMP cause problem of paucity, so there was a need of architecture to solve this problem. Net Patrol provides us that needed architecture. As the name suggests MA patrol or tours the network by migrating to the network nodes.

Component of Net Patrol are

(i) The Enterprise

(ii) The Manager Node

(iii)The Managed Node

The architecture of NetPatrol is shown in figure 5[9][3]. The manager node creates a pool MA ready for dispatch. In net patrol three types of MA are created –green, orange and purple for account monitoring, performance monitoring and root cause analysis. The agent acts between NM application and enterprise web services.

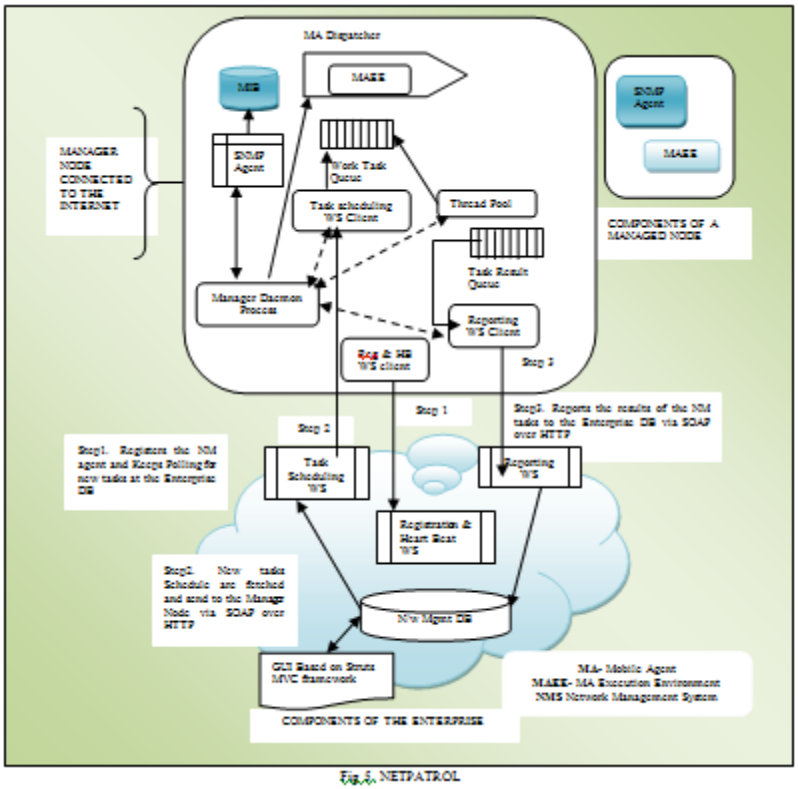

Fig. 5. NETPATROL

Components of **Manager Node** are

- A daemon process to bring agent
- MAEE to provide execution environment
- SNMP Agent

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

- MIB to store state of network parameters
- Result queue to maintain execution result

Net Patrol uses Service Oriented Enterprise. Thus it has three main web services that are **Registration and Licensing Web Service**, **Task Scheduling Web Service**, and **Result Reporting Web Services.**

In the manager node, daemon process will start up the Enterprise Registration and Licensing web service and register itself with a Network Monitoring id, which is unique in the entire enterprise database. The agent constantly polls the Enterprise Task Scheduler Web Service to check if there are any NM assign to it. Then the result of such task is sent back to enterprise by Result Reporting Web Service.

In managed nodes, there is MIB which is used to maintain state of network parameters. There is also set of library files to manage the node communication. The software agent analyses the data collected and put it into Result Reporting web service. The main purpose of Net Patrol is to combine traditional client server SNMP with Remote programming approach.

VII. NETLORD

To provide infrastructure-as-a-Service, there is a need of virtualized datacenters. These virtualized datacenters support multi tenancy and scaling at very low cost. There are various datacenter network architectures which support these features but have various drawbacks like

- They are very expansive to scale
- Limited support for multi-tenancy
- Complex architectures

To overcome these drawbacks, NetLord has been designed. It only requires small amount of offline, one time configuration. It provides tenants with simple and flexible network abstraction.

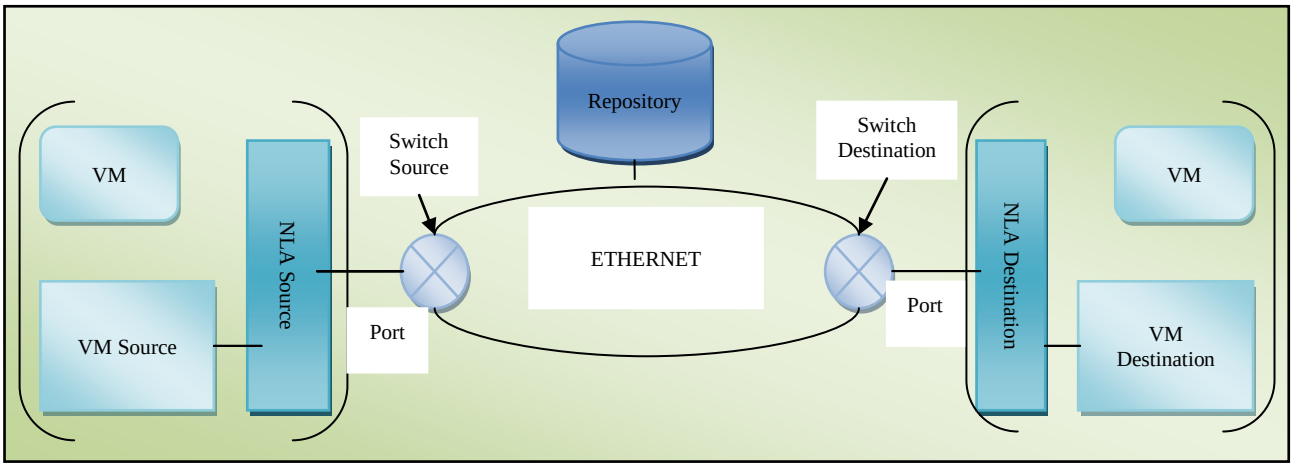

Fig 6. Architecture of NetLord

In NetLord'S design shown in figure 6[6] L2 packets of tenant are encapsulated and transferred to scalable L2 fabric. It also uses light weight agents so that this will be flexible architecture. Because of this method tenant virtual machine (VM) addresses are not exposed to hardware switches. This simplifies the operation and configuration of network hardware thus reducing the complexity.

Hence NetLord provides a very simple abstract view to tenant. The tenant will be having multiple private MAC address spaces in which he can assign arbitrary MAC addresses. NetLord provides full address space utilization, as same address can be used by multiple tenants without packets being misrouted.

Components of NetLord are-

Fabric switches- Switches should support VLANs and basic IP forwarding. They do not require support for complex routing protocols.

NetLord agents (NLA)-resides on the driver domain of each physical layer. Tasks of NLA are transparently encapsulating and decapsulating the packets and maintain all the information needed for encapsulation.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

Configuration repository- It maintain all the databases like paths, per tenant configuration information etc.

NetLord provides full flexibility to its tenants in choosing their addresses by virtualizing tenants's L2 and L3 address spaces. It also facilitates pre tenant traffic management. So, by this way NetLord provides virtual datacenters and multi tenancy service to its users at very low cost.

VIII. CONCLUSION

In this paper we have discussed various types of architectures to implement distributed network management efficiently. Also we have compare these architectures so as to see that which one is better option and what are the defects that the present technologies are suffering. In CS architecture SNMP is used to process and communicate the data. Certainly this architecture suffers many disadvantages like server crashing or traffic problem. So CS with MA offers solution to the above mentioned problem. Here CS works additionally with MA to increase the performance and to provide better scalability. Also there is MAGENTA architecture which offer services like fault tolerance, traceability of agents, flexibility of location of code and system independence for distributed administration.

In NetPatrol, integration of traditional SNMP and intelligent MA has been implemented to perform network management task in a distributed manner. And finally there is NetLord that provides multi tenant cloud datacenters. Table 1 below shows the features supported by different architectures.

Table 1. Comparative Analysis of various Network Management Model

Design Parameters v/s Models	Scalability	Load Balancing	Fault Tolerance	Dynamic Adaptation	Integration Of New Services
Centralized NMS Architecture	Very Poor due to request/response traffic	Low as central manager is the only authority	No as impossible to manage broken link	NO	NO
PEER NMS Architecture	Better as easily expandable	Complicated due to number of connections	Yes, as can work even after failure of clients, servers, agents	NO	YES
MAGENTA	Better as location independent	Uses mobile network	Mobile Agents, performs its task even if link breaks	YES	YES
NETPATROL	Combination of Web Services and mobile agents	WS Clients: Vital for Enterprise-Agent SOAP message exchange.	Mobile Agents, performs its task even if link breaks	NO	YES
NETLORD	Use of MAC address	Multiple paths due to VLAN	Continuous monitoring of paths, detects failure and then provide reroute	NO	YES

REFERENCES

1. Akhil Sahai and Christine Morin, "Enabling a Mobile Network Manager(MNM) through mobile agents", INRIA-IRISA Campus de Beaulieu 35042, Rennes CEDEX, France, Page 249-260,1998.
2. Akhil Sahai and Christine Morin, "Towards Distributed and Dynamic Network Management", INRIA-IRISA Campus de Beaulieu 35042, Rennes CEDEX, France Page 455-464,Feb 1998.
3. Anish Saini, Atul Mishra, A.K. Sharma," Distributed Network Management Architectures: A Review" International Journal of Computer Applications (0975 – 8887) Volume 68– No.3, April 2013.
4. Bohoris, C., Liotta, A., Pavlou, G., 2000. "Evaluation of constrained mobility for programmability in network management". In: Proceedings of the 11th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM'2000), pp. 243–257, December 2000.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

5. Damianos Gavalas, George E.Tsekouras and Christos Anagnostopoulos "A mobile agent platform for distributed network and system management". Elsevier-The journal of Systems and software Department of Cultural Technology and Communication, University of the Aegean, Arionos and Sapfous Street, 81100 Mytilene, Lesvos Island, Greece,2008.
6. Jayaram Mudigonda, Praveen Yalagandula, Jeff Mogul, Bryan Stiekes, Yanick Pouffary, "NetLord: A Scalable Multi-Tenant Network Architecture for Virtualized Datacenters" Proceedings of ACM SIGCOMM conference Vol 41,Issue 4 Page 62-73,2011
7. J. E. White(1996), "Mobile agents," in Software Agents (J. Bradshaw, ed.), Menlo Park, CA: The MIT Press, pp. 437–472,1996.
8. Mani Subramanian,"Network Management: Principles and Practice", Pearson Education,2008.
9. Mydhili K.Nair and V.Gopalakrishna," Applying web services with mobile agents for computer network management", International Journal of Computer Networks & Communications (IJCNC) Vol.3, No.2, March 2011.
10. Puliafito, A., Tomarchio, O., Vita, L.,. "MAP: design and implementation of a mobile agent's platform". Journal of System Architecture 46 (2), Page145–162,2000.
11. Rajesh P, Ranjiith S., Soumya P.R, Karthik V, Datthathreya S,"Network management system using web services and service oriented architecture : A case study", Proceedings of IEEE Network Operations and Management Symposium(NOMS'06), Vancouver B C, 3-7 April, 2006.
12. W. Stallings. "SNMP, SNMPv2 and CMIP: The practical guide to network management standards" Addison-wesley publication, 1994.