An Analysis of Management Cost for Mobile Agent Based Network Management Model

Atul Mishra
YMCA University of Sci. &Tech.
Faridabad
Haryana, India

A.K. Sharma
YMCA University of Sci. &Tech.
Faridabad
Haryana, India

Ashok Madan
YMCA University of Sci. &Tech.
Faridabad
Haryana, India

ABSTRACT
Client/Server based network management models involve the transmission of large amount of management data towards the centralized management station for processing. The staleness of gathered data (due to network latency involved) and probable error in the selection of management task being carried over (owing to the staleness of data) reduces the reliability of such management applications. In this sense, use of mobile agents offer many possibilities for designing the next generation of distributed network management systems. This paper discusses various mobile agent based network management models, the key advantages of mobile agents in the distributed network management systems and presents a mathematical model for the purpose of comparing client/server vs. mobile agent paradigms in terms of responsiveness and traffic generated around management station. Further, the existing mobile agent based systems follow arbitrary itinerary wherein it is susceptible to travel on the links which might incur high bandwidth utilization & greater roundtrip time. In this work a minimum spanning tree is constructed for the network represented as undirected graph. The itinerary obtained thereof is followed by the MA agents resulting in reduced bandwidth utilization and shorter roundtrip time.

Keywords
Mobile agents, Network Management, Distributed, SNMP, Scalability.

1. INTRODUCTION
The activities involving, operation, administration, maintenance and provisioning of network resource and services, is called network management [1]. Today's network management models are mostly based on CMIP [2], SNMP [3] and similar client/server based management protocols and hence suffers from scalability and flexibility problems as it involve the transmission of large amount of management data towards the centralized management station for processing. In the event of network stress (e.g. generation of lots of fault data or performance monitoring data) it overloads the management station. It is further complicated by distributed, often mobile, data, resources, service access and control [4], especially when these networks are growing in size and complexity. Moreover, varied technologies, such as SONET, ATM, Ethernet [5], DWDM etc., present at different layers of the Access, Metro and Core (long haul) sections of the network, have contributed to the complexity in terms of their own framing and protocol structures. Thus, controlling and managing the traffic in these networks is a challenging task [6].

These problems have motivated a trend towards distributed management intelligence that represents a rational approach to overcome the limitations of centralized NM. As a result, several distributed management frameworks have been proposed both by researchers and standardization bodies [7] [8]. However, these models are typically identified by static management components that cannot adapt to the evolving nature of today's networks, with rapidly changing traffic patterns and topology structures.

Of-late, the Mobile Agent (MA) paradigm has emerged within the distributed computing field. The term MA refers to autonomous programs with the ability to move from host to host to resume or restart their execution and act on behalf of users towards the completion of a given task. One of the most popular topics in MA research community has been distributed NM [9][10][11], wherein MAs have been proposed as a means to balance the burden associated with the processing of management data and decrease the traffic associated with their transfers (data can be filtered at the source).

The independence and mobility of mobile agents reduce client server bandwidth problems by moving a query from client to the server. It not only saves repetitive request/response handshake but also addresses the much needed problems created by intermittent or unreliable network connections. Agents can easily work off-line and communicate their results when the application is back on-line. Moreover, agents support parallel execution (load balancing) of large computation which can be easily divided among various computational resources.

1.1 Conventional Network Management System
A typical organization model of a network management system is based on SNMP Client/Server architecture. It consists of two major components: network agent process and the network manager process. The network agent process resides on the managed network devices such as routers, switches, servers etc. The network manager is housed on the NMS station from where it manages the various devices, by accessing the management information, through the agents residing on them as shown in Figure 1. The management information consists of collection of managed objects, stored in Management Information Base (MIB).

Various management applications such as configuration management, fault management etc. resides on the NMS stations whereas manager/agent paradigm procures the needed data for respective management application.

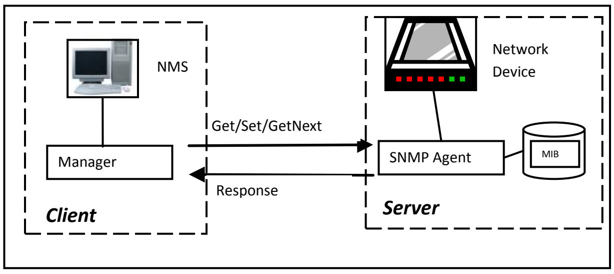

Figure 1: SNMP Client/Server Architecture

The agents have very simple interfaces by means of which they provide information to the requesting applications on granular basis gathered from the target devices. As they lack the needed intelligence and global view, agents don't perform management actions on their local data. Management data has to be transported to managers for taking any management decision. The management protocols provide the primitives for exchanging the management information among the agents and managers. Inherently it leads to centralized model of network management.

1.2 Limitations of Client/Server Based centralised Network Management Models

A centralized architecture suffers from the lack of scalability and flexibility. Furthermore, the staleness of gathered data (due to network latency involved) and probable error in the selection of management task being carried over (owing to the staleness of data) reduces the reliability of the management applications.

The following are major issues of client server based centralised Network management models.

1. **Centralized Management:** In this model, network manger plays a role of a centralized control unit. All the management decisions are taken by a single network node. As the network size grows so the efficiency of network management decreases. One of the drawbacks of centralized management is that if the management node fails, the overall network management would fail.
2. **Scalability:** All the management data is transported to management station for management decision. This doesn't scale as the network grows in complexity and size.
3. **Bandwidth wastage:** In Client/Server model, the bandwidth usage associated with management traffic increases as level of hierarchy increases. Thus a large amount of network bandwidth is consumed by network management operations in Client/Server model.
4. **Response Time:** Major The response time of a request depends upon the number of hop count between manager and managed device. In Client/Server model, the response time increase as level of hierarchy increases.
5. **Fault Tolerance:** The fault tolerance capability of Client-Server NMS is least or zero. This is one of the major draw back of the Client/server based network management models.

These problems have motivated a trend towards distributed management intelligence that represents a rational approach to overcome the limitations of centralized NM. Gathering and analysis of the management data from agents of managed devices is partitioned and spread over the various computing platforms in the network (sometime the managed devices act as computing platform) thereby breaking the centralized paradigm. Code mobility [12] offers an attractive alternative to centralized architectures.

1.3 Distributed Management based on mobile code

The use of approaches based on mobile code for network management allows overcoming some limitations of current centralized management systems [13]. Mobile code mobility includes different technologies, all sharing a single idea to enhance flexibility by dynamically transferring programs to distributed devices and have these programs executed by the devices. The program transfer and execution can be triggered by the service itself, or by an external entity. In particular, they identified three different types of mobile code paradigms

- Code on Demand
- Remote Evaluation
- Mobile Agents.

1.3.1 Code on demand Paradigm

In Code on demand paradigm, Client is able to access the required resource which is located at same place. However, it lacks the information (code) on how to process such resources. Thus Client interacts with the server, requesting the service know-how. A second interaction takes place when server delivers the know-how to client which can subsequently execute it as shown in Figure 2.

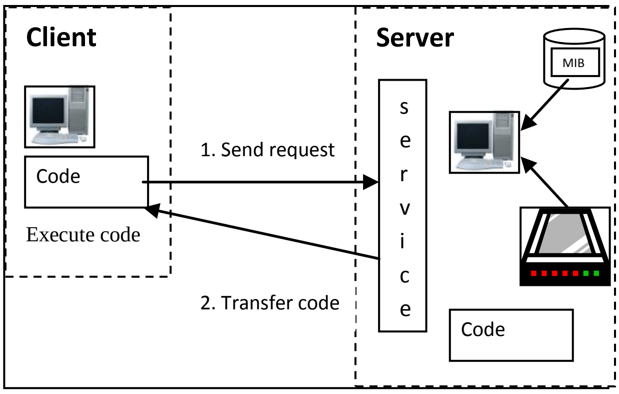

Figure 2: Code on Demand Paradigm

1.3.2 Remote Evaluation paradigm

In the Remote Evaluation paradigm, a client has the know-how necessary to perform a service but it lacks the required resources, which are located at a remote server.

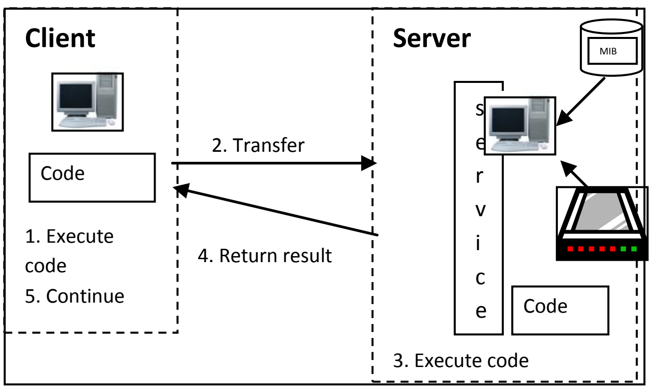

Figure 3: Remote Evaluation Paradigm

Consequently the client sends the services know-how to the remote site that executes the code using the resources available there as shown in Figure 3.

1.3.3 Mobile Agent Paradigm

In MA paradigm, the service know-how is owned by the client, but some of the required resources and data are located at a remote server. Hence the component migrates to the server carrying the know-how and possibly some intermediate results. After its arrival, A completes the service using the resource available. Mobile agent System (MAS) resides on the server and provides the needed environment for the mobile agent to run on the server as shown in Figure 4.

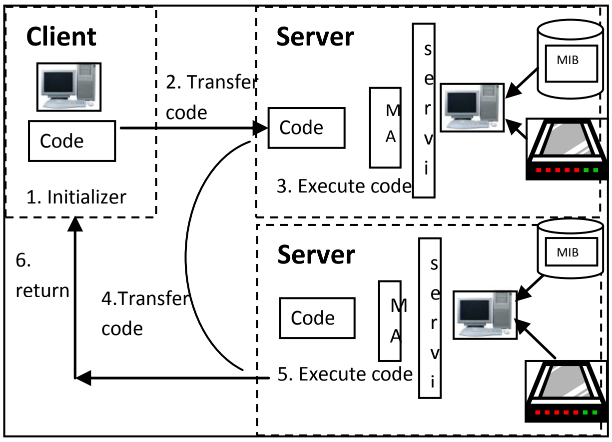

Figure 4: Mobile Agent Paradigm

Mobile agents allow more and more of the network management intelligence to move closer to the devices unlike the centralized model. Some micromanagement operations could be performed locally avoiding the need to transfer large amount of data generated at management nodes to the central management station thereby reducing the workload for the management station and the overhead in the network.

2. RELATED LITERATURE

Early work in the field of mobile code, carried over by Goldszmidt et al [14], introduces the concept of management by delegation. Herein, the management station can extend the capability of the agents at runtime thereby invoking new services and dynamically extending the ones present in the agent on the device. Mobile agent based strategies have distinct advantages over the others as it allowed for easy programmability of remote nodes by migrating and transferring functionality wherever it is required. Bellavista et al. [15] proposed a secure and open mobile agent environment, MAMAS (Mobile Agents for the Management of Applications and Systems) for the management of networks, services and systems. Stephan et al. [16] introduce the concept of mobile agent based network management platform, where in a location independent network manager and assists the administrator to remotely control his/her managed network, through launching MAs to carry out distributed management tasks. In [17], I. Satoh proposes how a network and application independent MA based framework could be designed. Manoj Kumar Kona et al. [18] described an SNMP based efficient mobile agent network management structure, in order to cooperate with conventional management system; For transferring less network monitoring data and managing devices more effectively, Damianos Gavalas et al.[9] propose a scalable and flexible MA based platform for network management; Chi-Yu Huang et al. proposed a clustering mobile agent based network management model aiming at large enterprise entrant network in [19]. Liotta et al. [20] have suggested an MA-based hierarchical and dynamic management architecture which deploys static middle managers who in turn can launch MAs. Pualiafito et al. [21] introduce the Mobile Agent Platform (MAP), used for monitoring the systems state by calculating aggregation functions combining several MIB values (*health* functions). Damianos Gavalas et al.[22] proposed a hierarchical and scalable management model where middle managers are themselves mobile and based on certain policies they dynamically segment the network and deploy other mobile middle managers for data collection.

2.1 Mobile agents based network management model

Damianos Gavalas et al. [22] have discussed number of mobile agent-based network management models in their research work. Few important ones have been discussed and compared against certain design parameters in this section.

2.1.1 Flat bed Model

For a particular management task a single mobile agent is launched from a management station which then traverses the network topology in a sequential manner, visiting each managed device and carrying out the assigned task.

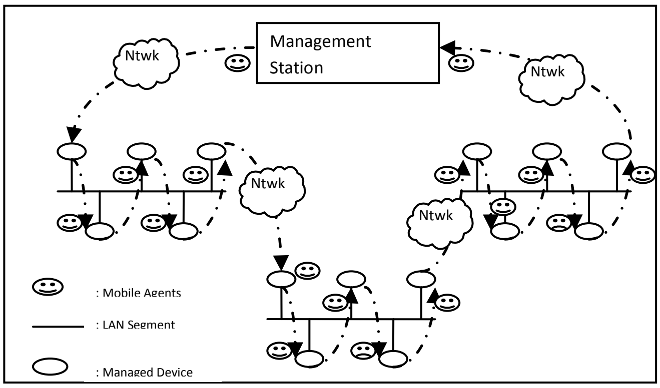

Figure 5: Flat Bed Model

Though the model relives the network from the flood of request/response messages, it introduces the issue of roundtrip delays as the network size grows. This leads to scalability issue if data has to be collected very frequently from managed devices. The size of MAs grows considerably in large networks. The model is shown in Figure 5

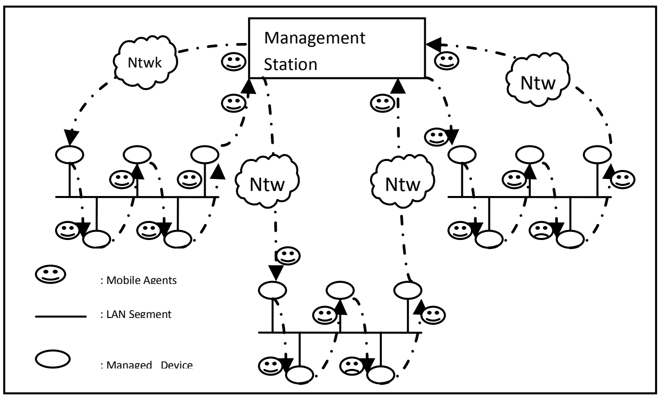

Figure 6: Segmentation Model

2.1.2 Segmentation Model

Here the scalability issue is addressed by partitioning the network into many administrative or geographical domains and assigning a single MA entity to each one of them. This brings high degree of parallelism in the data collection architecture and brings the response time down by many folds. The model is shown in Figure 6

2.1.3 Hierarchical Static middle manager model

The scalability problem is more adequately addressed by deploying hierarchical models wherein NM tasks are delegated to MAs. They migrate to remote subnetworks/domains where they act as local managers and takes over the responsibility of local devices from the central manager. These models suffer from automatic adaptation of management system to changing network configurations, i.e. mid level manager do not change the location where they execute. The model is shown in Figure 7.

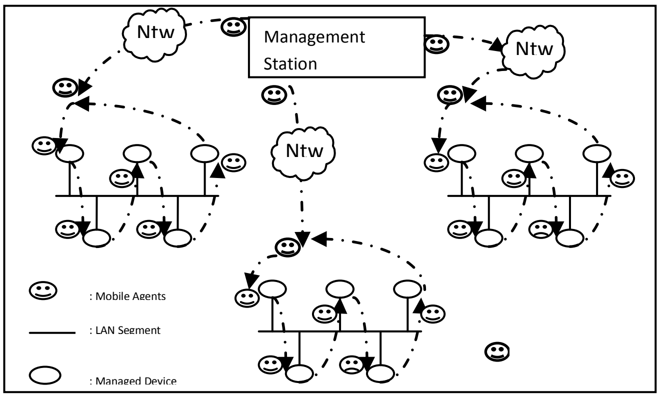

Figure 7: Hierarchical *Static middle manager* model

2.1.4 Hierarchical Mobile Middle manager model

In search of more flexible solutions, a concept of Mobile Middle Manager (MDM), referring to a management component that operates at an intermediary level between the manager and the management end points, is introduced. The mobility feature of the MDMs allows the management system to adapt dynamically to a changing network conditions. MDMs can be deployed to or removed from a given network segment in response to change in network traffic or move to a least loaded host to optimize local resource usage. The model is shown in Figure 8.

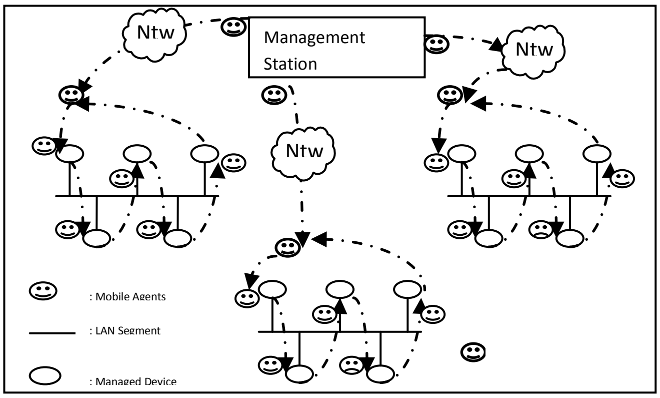

Figure 8: Hierarchical *Mobile middle manager* model

3. ROLE OF MOBILE AGENTS IN DISTRIBUTION OF MANAGEMENT TASKS

3.1 Mobile Agent Technology

Mobile agents were introduced in the early 90's within the artificial intelligence research community, as semi-intelligent computer programs that assist a user with large amounts of complex information within a network environment. These are typically dispatched from one node in a network and transported to a remote node for execution. This concept of remote programming using mobile agents is considered as an alternative to the traditional client-server programming based on the remote procedure call or the static distributed object paradigm (e.g. CORBA). The primary goal of using mobile agents in management of telecommunication network is reducing network traffic by using load balancing and building scalable and reliable distributed network management system.

3.2 Distribution of Network Management Tasks

Typical Network management systems have to deal with various functional areas in order to ensure availability of resources and services of the network to its end users. SNMP and CMIP require gathering and analyzing large amount of data from the network for various management tasks. SNMP manages and monitors only network elements and agents provide a limited and fixed set of functions [23]. Existing management models traditionally adopt a centralized, client/server (CS) approach wherein the management application (with the help of a manager), acting as clients, periodically accesses the data collected by a set of software modules (agents) placed on network devices by using an appropriate protocol. These systems regularly suffer from poor scalability due to an increase in the amount of communication and generate too much traffic in the network and the number of failures in nodes and channel.

Therefore, there is a need to employ mobile agents as an autonomous entity in network management and transfer the administration tasks to them. Also under this situation the network management tasks and computational load are distributed instead of being centralized towards and on the manager host. One of the important goals of the network management is to have balanced loading and reliable loading on the network such that connections in the network can be established quickly without noise, or several trails. Network management also aims to organize the networks in order to work professionally, successfully adjust to changes, and react to problems such as traffic patterns.

The OSI management model recognizes the following important network management functional areas. These are: Fault Management, Accounting Management, Configuration Management, Performance Management and Security Management. Besides these networks topology discovery is one of the fundamental functions of the management systems. Typically it involves finding the devices of the network and interconnection among those. A more detailed implementation would focus on construction of more detailed views that may include, for example, services available on a device or devices that satisfy certain constraints. If the constraints are functions of device status, then we approach problem discovery. As the complexity of discovery grows, it is harder to implement using classical client/server approaches.

The same principles as we saw in network modeling can be used to diagnose network faults. Detection of faults is a process of building a specialized model of the network. For example, a simple agent performing selective discovery of nodes with utilization that exceeds a certain threshold builds a model of over-utilized nodes. If the constraints on discovery describe violations of what is considered normal behavior of network elements, then the agents testing the constraints perform a fault detection function.

Another area of importance is performance measurement, which involves gathering statistical information about network traffic, methods to reduce, and present data. Measuring performance of networks using centralized SNMP based management is very difficult due to reasons like network delays and information traffic jam at the central management station.

It is now widely recognized that the use of decentralization in this kind of applications potentially solves most of the problems that exist in centralized client/server solutions. Hence applications can be more scalable, more robust, can be easily upgraded or customized and they reduce the traffic in the network.

In a distributed network, the network operator monitors the trend of network flow to assess network performance and identify unusual conditions. The analysis of data can be achieved from the management information base. The management information base preserves various data objects for network management. The information in management information base is ordered in clusters and maintained in a tree-like structure. Thus management information base manage the complex network tasks in the distributed network management environment.

4. MODEL FOR EVALUATION OF MOBILE AGENTS IN NETWOK MANAGEMENT SERVICE

As discussed above, the key entities involved in network management model are as follows:

- Manager: This is the entity that knows how to execute the job according to its knowledge of the management activity. It is usually housed at NMS station.
- Agent**:** This is the entity that owns and provides raw data collected from the network elements. An example is the agent in SNMP protocol that handles the Management Information Base (MIB) and provides data to the Manager.
- Management Application: This is the entity that needs the result generated by the Manager. For instance, the management station has many management applications like configuration management, fault management etc. which need data provided by the manager.

As shown in Figure 9, at the beginning of the job, the manager needs to access the data stored in the MIB at node N1. After the interactions between the manager and N1 are completed, manager may need to access the information at another node N2 and repeat the data access procedure described below.

Algorithm shown in Figure 10 captures the essence of the interaction between a central manager residing on a management station and various agents residing on the remote managed devices. For any management activity, say collection of certain performance monitoring parameters, certain jobs are defined and manager runs through various jobs in succession to accomplish the management activity.

It may be noted that, according to different management jobs, this data accessing interaction between the manager and agents may repeat many times during the management process.

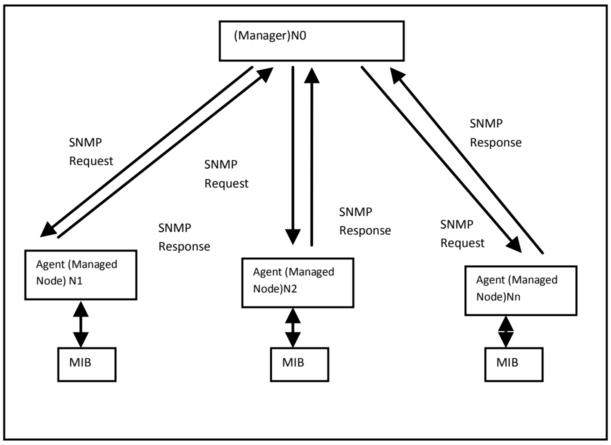

Figure 9: Management Entity and Interaction Model

Algorithm dataaccess()

1. Start job J;
2. i=1;
3. while (i<N_{n+1}){

3.1. Access MIB at node N_i to get data;

3.2. Process data to perform the relevant job;

3.3. Repeat steps 3.1 to 3.4 till end of data from node Ni;

3.4 i++; // Move to next node N_{i+1}

}

1. Perform rest of the work;
2. close job J;

Figure 10: Algorithm of interaction model

During the management process, the manager may generate management results, and these results may be required to be reported to a central management activity. If the interaction happens on the same site, it is termed as a local interaction, whereas an interaction between two entities associated with two different networks is termed as a remote interaction over the network, thereby affecting the management performance of the deployed management models.

4.1 Performance Metrics and parameters of client/server and MA paradigm

In this work the following two performance matrices have been developed with a view to compare the performance of Client/Server and MA models of network management.

4.1.1 Network Traffic related performance

The following traffic related performance parameters illustrate the overhead introduced by the management application and paradigm of the network.

- Traffic generated around central manager residing on the NMS,
- Total management traffic generated in the network.

The above mentioned parameters can also be used to determine the potential bandwidth bottlenecks.

4.1.2 Time Related performance

The following time related performance parameters, such as

- Total time taken by typical management activities.
- The remote interactions time between entities on different nodes in the network.

4.2 Analysis in Client/Server Paradigm

Consider the Figure 9. Let us say Sreq is the size (bytes) of SNMP request initiated by the Manager at node N0 in Client/Server paradigm and Sres is the response data size (bytes) accessed by the manager from node Ni (i = 1…n) which include data collected from MIB.

Considering the algorithm stated above, the average traffic around the Manager at node N0 can be computed as given below

$$C_{cs}^{m} = \sum_{i=1}^{i=n}\{(S_{req} + S_{res})\} * p \quad (1)$$

Where C_{cs}^{m}: The management cost for Client/Server paradigm in terms of traffic generated around a particular (here N0) network device and

p: the number of times polling done.

Furthermore, if V_{mib} is number of MIB variables accessed at each node

$$C_{cs}^{m} = \sum_{i=1}^{i=n}\{(S_{req} + S_{res}) * V_{mib}\} * p \quad (2)$$

Based on Equation (2), the total execution time taken by a central manager in Client/Server paradigm to complete a job

$$T_{cs}^{t} = \sum_{i=1}^{i=n}\{(((S_{req} + S_{res})/B_{wi}) + 2L_{ti} + t_a + t_p) * V_{mib}\} \quad (3)$$

Where

T_{cs}^{t}: Total execution time (seconds) in CS paradigm.

Bwi: Bandwidth (bps) of the link btw nodes N0 and Ni.

Lti: Latency (seconds) between nodes N0 and Ni.

t_a: Average time (seconds) for the MIB access on a given node.

t_p: Average time (seconds) for processing of data at the central node.

By removing the average local interaction time, i.e. t_a and t_p from the total execution time of a job, the average remote interaction time is computed as below.

$$T_{cs}^{r} = \sum_{i=1}^{i=n}\{(((S_{req} + S_{res})/B_{wi}) + 2L_{ti}) * V_{mib}\} \quad (4)$$

Where

T_{cs}^{r}: Remote interaction time (seconds) in CS paradigm.

4.3 Analysis in Mobile Agent paradigm

In mobile agent model, as shown in Fig. 11, a mobile agent containing the necessary logic for the required management activity is dispatched to a remote node. Instead of moving all the raw data from the node N1 to the manager to perform the computation, the MA applies the algorithm on the data locally and only carries the partial result at the next node. Finally an overall report derived from the partial results from all the nodes is taken back to the central manager.

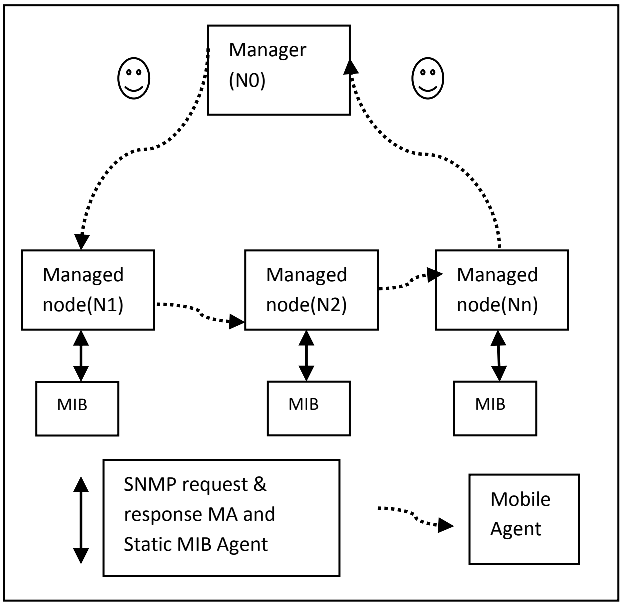

Figure 11: Mobile Agent interaction model

Therefore the average traffic around the Manager at node N0 can be computed as given below

$$C_{ma}^{m} = \{Sma + \sum_{i=1}^{i=n}(Spr)\} \quad (5)$$

Where

C_{ma}^{m}: Management cost for MA paradigm in terms of traffic generated around a particular (here N0) network device,

Sma: The size (bytes) of the MA,

Spr: The size (bytes) of intermediate partial result generated in MA paradigm at each node.

The total execution time of job

$$T_{ma}^{t} = \sum_{i=1}^{i=n}\{((S_{ma} + S_{pr})/B_{w(i-1,i)})) + L_{t(i-1,i)} + (t_a + t_p) * V_{mib}\} \quad (6)$$

Further, the average remote interaction time

$$T_{ma}^{r} = \sum_{i=1}^{i=n}\{((S_{ma} + S_{pr})/B_{w(i-1,i)})) + L_{t(i-1,i)}\} \quad (7)$$

Where

T_{ma}^{t} : Total execution time (seconds) in MA paradigm.

T_{ma}^{r} : Remote interaction time (seconds) in MA paradigm.

By removing the average local interaction time, i.e. t_a and t_p from the total execution time of a job, the average remote interaction time is computed.

4.4 Client/Server paradigm v/s Mobile Agent paradigm

In this section we compare the corresponding performances of the client server and the mobile agent paradigm. Based on the expressions obtained in the previous section we conclude as captured in Table 1.

Table 1. Comparison of client/server v/s MA model

Performance Matrix	Client/Server Model	Mobile Agent Model
C^m (management cost in terms of traffic generated around a particular network device)	This is proportional to the number of nodes (MIB) accessed by the manager and number of times a particular node (MIB) accessed.	It is proportional to the size of results in bytes collected from various nodes.
T^r (remote interaction time (seconds) in paradigm)	This is proportional to time taken for remote interactions done to access nodes (MIBs) and increases with the increase in number of MIBs	As interaction is local between the manager (MA) and managed device, it doesn't increase with the increase in number of MIBs.

Consider Equation (2) and (5) wherein it may be noted that given the size of the MA (Sma) is negligible as compared to the total network management traffic, the traffic (Spr) generated around the central management station (N0) in MA based NMS is far less than that of traffic (Sreq + Sres) in Client/Server based NMS. The point to note is that Spr is only a partial report consisting of a very small amount of pre-processed data generated by the MA on the host node. On the contrary in the CS scenario a comparatively a large amount of data migration is done from the device to the central manager and the processing is done centrally. As Spr is much less in size than (Sreq + Sres) processed by the CS model, thus, Ccs >>>> Cma. Thus it is clear from the presented analysis that the MA based NMS is better than Client/Server based NMS.

For theoretical quantitative evaluation of management cost for two network management models let us assume the following:

SNMP request packet size (Sreq) = 50 Bytes,
Sma (MA size) is 3 KB =1024*3 = 3072 Bytes,
Data accessed by the task manager (Sres) in CS paradigm = α times of Sma (Size of mobile agent << raw data collected in Client/Server model),
Sr (Partial Result) in MA paradigm= 200 bytes.

The management cost in terms of flow of management traffic around the management station in CS paradigm for a typical node is computed as follows.

Ccs= (Sreq+Sres) = 50+ α * Sma

Putting parameters in Equation (1) the management cost (Ccs) for the values of α = 5 & 30 for a typical node is computed below.

Case A: Taking α =5,

Ccs = 50+ 5 * 3072

= 50 + 15360

= 15410 Bytes

Case B: Taking α =30,

Taking α =30

Ccs = 50+ 30 * 3072

= 50+92160

= 92210 Bytes

Putting the parameters in Equation (5) the management cost in MA Paradigm for a typical node is computed below.

Cma = (Sma + Sr)

= (3072+200)

= 3272 Bytes

Table 2. Traffic around Management Station in C/S vs MA

No. of Nodes	MA	CSA=5	CSB=30
1	3272	15410	92210
5	4072	77050	461050
10	5072	154100	922100
20	7072	308200	1844200
50	13072	770500	4610500

It may be noted from Table 2 that with the increase in number of nodes the management cost of C/S increases many fold as compared to MA model as also illustrated in Fig. 12.

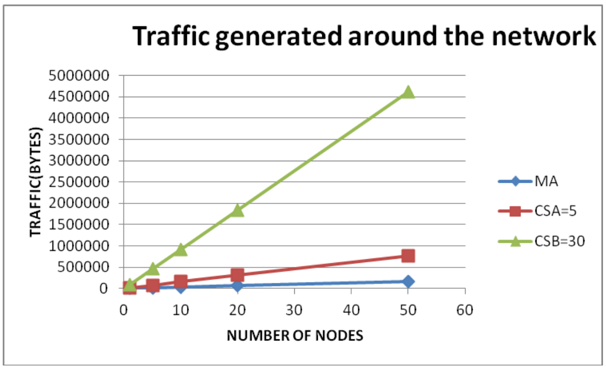

Figure 12: Traffic around management station

5. SPANNING TREE BASED TRAVERSAL PLANNING FOR MA BASED NETWORK MODELS

As discussed above, the MA based NMS is better than **CS** based NMS. But the itinerary planning for MA based NMS can further improve the bandwidth utilization.

When MA follows the arbitrary route, it is very likely that it traverses on the link having a high bandwidth utilization and greater round trip time. Thereby it increases the overall cost of bandwidth utilization. Whereas a carefully planned itinerary for MA based network management can improve the bandwidth utilization and round trip time.

In this work a spanning tree based mechanism is being proposed that improves the itinerary for mobile agents in MA based network management model. The working of the proposed technique is given below.

- Here, a network is represented as an undirected and connected graph, say G, consisting of nodes (V) and edges (E) representing managed devices and communication links respectively, which can be represented as G = (V,E)

- In order to discover an MST (minimum spanning tree) for G, Prim's Algorithm is applied, which is explained briefly here.

Algorithm MST(G)

1. Choose any element r; set S = {r} and A = 0 (Take r as the root of spanning tree, S as the set of vertices in MST and A as the set of edges in MST)

2. Find the lightest edge, in terms of weight given to an edge (here, bandwidth of the link), such that one point is in S and the other is in V\S (all the vertices not in S). Add this edge to A and its (other) endpoint to S.

3. If V\S = 0, then stop and output the minimum spanning tree (S,A).

Otherwise, goto step 1.

Let us take an example to discuss the algorithm explained above. Consider an undirected weighted graph having a 7 nodes (V1 to V7) and MA starts traversing from node V1 and traverse all the nodes (V2 to V7) to collect data from each node as shown in Fig. 13.

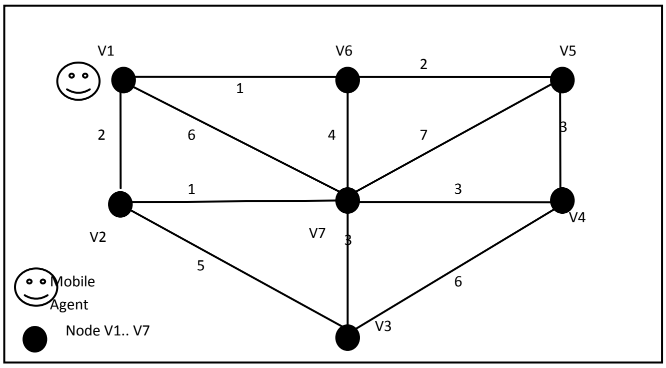

Figure 13: Mobile Agent Network represented as undirected weighted graph

Along the edges the weights (cost coefficient) shown essentially indicate the traversal cost which is indirectly proportional to bandwidth of the edge/link. The cost for network management does not only depends on management data size but also depends on cost coefficient of network link through which management data pass. To calculate the cost of the MA traversal all through the network assigned to it, following cost calculation considerations were carried out.

5.1 Itinerary for Mobile Agents

5.1.1 When MA follows Arbitrary Itinerary:

The MA start traversing graph from node V1 and follow the path from V1 to V7, V7 to V6, V6 to V5,V5 to V4, V4 to V3 and V3 to V2 as shown in Fig. 14. The weight is given as V1 to V7 =6, V7 to V6=4, V6 to V5 =2, V5 to V4 = 3, V4 to V3=5 and V3 to V2=5 respectively. The total cost coefficient of path is calculated as **(6+4+2+3+6+5=26)** as shown in Fig. 14.

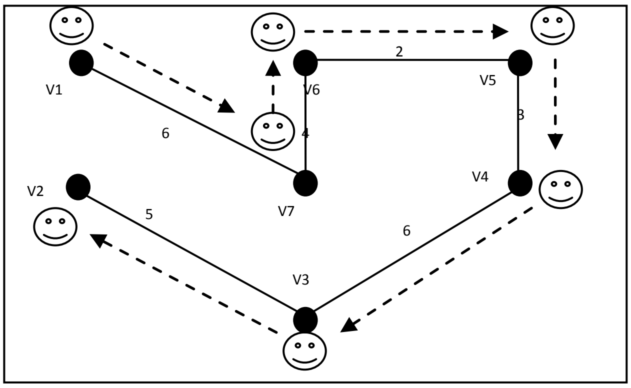

Figure 14: Path followed by Mobile Agent for arbitrary itinerary

5.1.2 When MA follows MST Based itinerary:

When the MA follow MST based itinerary as shown in figure xxx. The MA start traversing graph from node V1 and follow the path from V1 to V6, V6 to V5, V5 to V4,V4 to V7, V7 to V2 and V2 to V3 as shown in Fig. 15. The weight is given as V1 to V7 =1, V7 to V6=2, V6 to V5 =3, V5 to V4 = 3, V4 to V3=1 and V3 to V2=5 respectively.

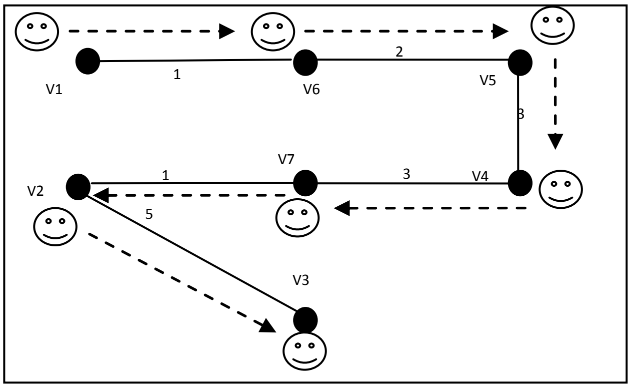

Figure 15: Path followed by Mobile Agent for MST itinerary

The total cost coefficient of path is calculated is **(1+2+3+3+1+5=15)** as shown in Fig. 15. It is clear that MST based itinerary is better than arbitrary itinerary.

The adjacency matrix of undirected weighted graph is given in Fig. 16.

The adjacency matrix of weighted graph is given as input to the algorithm. The output of the algorithm returns the adjacency matrix to a minimum spanning tree.

	V1	V2	V3	V4	V5	V6	V7
V1	-	2	-	-	-	1	6
V2	2	-	5	-	-	-	1
V3	-	5	-	6	-	-	3
V4	-	-	6	-	3	-	3
V5	-	-	-	3	-	2	7
V6	1	-	-	-	2	-	4
V7	6	1	3	3	7	4	-

Figure 16: Adjacency Matrix of undirected weighted graph

From cost computations done in earlier, Cma=3272 Bytes

Case A: Non MST (Arbitrary) Route, Total Cost Coefficient = (6+4+2+3+6+5=26)

Management Cost Cma= 3272*26 =85072 Bytes

Case B: MST Based Route, Total Cost Coefficient = (1+2+3+3+1+5=15)

Management Cost Cma= 3272*15 = 49080 Bytes

Table 3. Management Cost of Non MST v/s MST based traversal

Number of Nodes	A(Non-MST)	B(MST)
1	85072	49080
5	425360	245400
10	850720	490800
20	1701440	981600
50	4253600	2454000

It may be noted from Table 3 that with the increase in number of nodes the management cost of non-MST based traversal increases many fold as compared to MST based traversal model as also illustrated in Fig. 17.

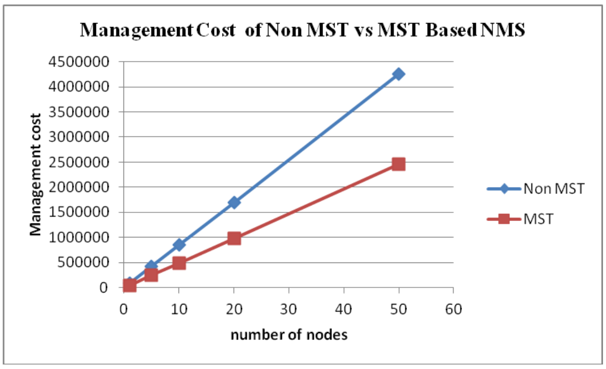

Figure 17: Management Cost Comparison for different traversal scheme

6. CONCLUSION

Mobile agents offer an easy re-configurable, flexible and scalable solution to the management of today's complex telecommunication networks thereby reduces the number of necessary human interactions. Many of the complex management tasks can be delegated to agents whom agents can easily carry out without much intervention from the higher management layers. As discussed in the mathematical model analysis, the independence and mobility of mobile agents reduce bandwidth overloading problems by moving a processing of the management data and decision making from centralized management stations to the managed devices thereby saving many repetitive request/response roundtrips and also address the problems created by intermittent or unreliable network connections between the network management stations and managed devices. Agents can easily work off-line and communicate their results when the application is back on-line. Moreover agents support parallel execution (load balancing) of large computation which can be easily divided among various computational resources. Thus using agents network monitoring and other management tasks can be easily decentralized.

7. REFERENCES

[1] Mani Subramanian, "Network Management: Principles and Practice", Pearson Education, 2008 Edition.

[2] Y. Yemini, "The OSI Network Management Model", IEEE Communications Magazine, vol.31, no.5, pp.20-29. May 1993.

[3] W. Stallings, SNMP, SNMPv2, SNMPv3 and RMON 1 and 2, third ed. Addison Wesley, 1999.

[4] S. Makki. and S. Wunnava, "Next Generation Networks and Code Mobility." ISAST Transactions on Communication and Networking, No.1, Vol 2, pp 42-47, March, 2008.

[5] H. Chamas, W. Bjorkman, and M.A. Ali, "A Novel Admission Control Scheme for Ethernet Services",Proceeding of IEEE ICC 2005.

[6] M. D'Arienzo, A. Pescape, and G. Ventre, "Dynamic Service Management in Heterogeneous Networks", JNSM: Vol. 12, No.3, 2004.

[7] SNMP research, "The Mid-Level Manager", http://www.snmp.com/products/mlm.html

[8] G. Goldszmidt, Y. Yemini and S. Yemini, "Network management by delegation", Proceedings of the 2nd Int. Symposium on Integrated Network Management (ISINM'91), April,1991.

[9] D. Gavalas, G.E. Tsekouras, and C. Anagnostopoulos, "A mobile agent platform for distributed network and system management", Journal of Systems and Software, Vol 82, pp. 355-371, 2009

[10] Mydhili K Nair and V. Gopalakrishna, "Applying Web Services with Mobile Agents For Computer Network Management", International Journal of Computer Networks and Communications (IJCNC), Vol. 3, No. 2, pp 125-144, March, 2011.

[11] S.S. Manvi and P. Venkataram, "Agent based subsystem for multimedia communications". IEEE Proceedings Software Journal 153 (1), 38–48, 2006.

[12] A. Fuggeta, G. P. Picco, and G. Vigna, "Understanding Code Mobility", IEEE Transactions on Software Engineering, Vol. 24, No. 5, pp. 436-361, 1998.

[13] M. Baldi, S. Gai, and G. P. Picco, "Exploiting Code Mobility in Decentralized and Flexible Network Management, Proceedings of the 1st International Workshop on Mobile Agents (MA'97), LNCS Vol. 1219, pp. 13-26, 1997.

[14] G. Goldszmidt, "Distributed Management by Delegation", In Proceedings of the 15th International conference on Distributed Computing Systems, 1995.

[15] P. Bellavista, A. Corradi and C. STefanelli, "An open secure mobile agent framework for systems management." Journal of Network and System management, 7:323-339, 2003.

[16] R. Stephan, P. Ray, and N. Paramesh, "Network Management Platform Based on Mobile Agents", International Journal of Network Management, Vol. 14, pp. 59-73, 2004

[17] I. Satoh, "Building and selecting mobile agents for network management.", International Journal of Network and Systems Management 14 (1), 147–169. March, 2006

[18] M. K. Kona and C. Z. Xu, "A Framework for Network Management Using Mobile Agents", Parallel and Distributed Processing Symposium, Proceedings International, pp. 227-234, IPDPS 2002.

[19] Chi-Yu Huang and C. Pattinson, "Using Mobile Agent Techniques for Distributed Manufacturing Network Management", PGNet 2001, Liverpool, UK.

[20] A. Liotta, G. Knight and G. Pavlou, "Modelling Network and System Monitoring Over the Internet with Mobile Agents", Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'98), pp.303-312, February 1998.

[21] Pualiafito A., Tomarchio O., Vita L., "MAP: Design and Implementation of a Mobile Agents and Platform",Journal of Systems Architecture, 46(2), pp.145-162, January 2000

[22] D. Gavalas, D. Greenwood, M. Ghanbari and M. O'Mahony, "Hierarchical network management: A scalable and dynamic mobile agent-based approach." Computer Networks, 38: 693-711, 2002.

[23] A. Tripathi, T. Ahmed and S. Pathak, "Paradigms for Mobile Agent Based Active Monitoring Of Network Systems", IEEE, 2002.owman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. .