Chapter 1

INTRODUCTION

1.1 NETWORK MANAGEMENT SYSTEM

Data communication is the need of current information technology. It has been evolving very rapidly since the earliest days of computing. Telecommunication industry and big enterprises are wholly dependent upon networked system for serving their information retrieval needs. A very reliable underlying communication and data infrastructure has become a key requirement for such enterprise environment. Faults and performance inefficiencies in these systems cause considerable business losses. Hence there is a need to manage network infrastructures, including communication elements, systems and applications.

Network management system is a fundamental facilitator that typically gathers and analyzes huge amount of data from the network and makes decisions thereof for various functional areas of a management system. The term Network Management (NM) [1] refers to the activities involving, operation, administration, maintenance and provisioning of network resource and services. The goal of network management is to ensure that the users of the network receive the information technology services with the quality of the service they expect.

A typical organization model of a network management system is based on SNMP [2] 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 1.1. 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. reside on the NMS stations whereas manager/agent procures for them the needed data.

Network management system typically gathers and analyzes huge amount of data from the network and makes decisions thereof for various functional areas of a management system. The real-time requirements for various types of traffic are also set instantly. 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.

The accumulation of functionality, intelligence and responsibilities at a central site represents a rigid management solution, it suffers from the well-known problems of centralised system which are lack of distribution, a low degree of flexibility and fault tolerance etc. [3]. The network management activities, dealing with data gathering and
reporting methods, involve substantial transmission of management data thereby consuming a lot of bandwidth and computational overhead. This leads to a considerable strain on the network and significant traffic jam at the manager host [4]. Besides this centralized management activities are limited in their capability as they cannot do intelligent processing like upfront judgment, forecasting, analyzing data and make positive efforts to maintain quality of service.

To address the above mentioned limitations of centralised, researchers started investigating decentralised, distributed solutions. However, the Simple Network Management Protocol (SNMP) which is the dominant protocol in the Internet management defined a centralised architecture.

Management distribution requirements were addressed up to some extent by hierarchical NMSs [5], which enable data pre-processing and delivery of higher-level information to the manager station. Most of the time in such models, a part of the management functionality is assigned to certain mid-level-managers. These selection and assignment of these managers is done at design phase and it is mostly static in nature. Hence, it cannot cope well with the high dynamics of modern networks, referring to frequent changes that need to be incorporated in the NMSs as well. Those changes may vary from the introduction of new devices and software to the provision of new types of services.

The requirement for flexibility combined with that for decentralisation has been addressed by the pioneering work of Goldszmidt et al. who introduced the concept of Management by Delegation (MbD) [28]. Delegation of management functionality refers to moving part of the intelligence traditionally residing on a central management station to distributed management entities running on selected managed systems. This allows the dynamic change of responsibilities and tasks allocation and leads to flexible systems with dynamically augmented capabilities. A main objective of MbD is to perform decentralised tasks in proximity to system resources. It prevents unnecessary data transfers and therefore eases the introduction of new tasks at runtime.
Nevertheless, approaches based on the code mobility paradigm cover broader aspects of management delegation and can include MbD as a special case. Code mobility refers to the capability of dynamically moving the components of a distributed application among the nodes of a network. This idea can be applied in the generic field of distributed systems however distributed management represents a very promising application domain, as it can benefit from the efficient use of bandwidth resources and the high degree of flexibility and configurability offered by mobile code.

A sub-class of the code mobility paradigms, termed the Mobile Agent (MA) paradigm, has emerged in the mid-90’s. Briefly, MAs can be defined as “computational components identified by their code, persistent and execution state, with the ability to autonomously move from host to host, acting on behalf of a user or an application”. In the distributed management domain, MAs add a new dimension by exploiting their ability to autonomously migrate to different network devices and perform complex management tasks without the manager’s intervention. The migration and autonomy features of MAs attracted the attention of many researchers and, as result, MA-based distributed management has become a very active research topic in the past few years.

**1.2 ROLE OF MOBILE AGENTS IN NETWORK MANAGEMENT**

MA technology has been very popular in the field of distributed computing field. Several research activities investigated its applicability in numerous application domains, including information retrieval, e-commerce, mobile computing and parallel computation. As a result of the intense research on MA technology, design of various mobile agent platforms has taken place over the last few years. Some of the known platforms are like Aglet [51], voyager [53], JADE [55] etc. Many of these platforms are specially designed for management applications, aiming at optimising flexibility and performance aspects which is missing in general-purpose platforms like Voyager, Grasshopper [54] etc.
These mobile agent platforms have been used as the basis for covering a wide spectrum of management applications, including network routing, network monitoring, fault, configuration, performance, security and service management areas. These applications demonstrated the applicability of MAs on several management scenarios, exploiting their migration and autonomy features to dynamically customise the management capabilities of network devices, perform semantic compression of data, or detect changes related to performance, faults or configuration of managed systems.

Though these approaches offer many advantages of MA-based design, however there are several limitations related to their practical use on distributed management. These limitations relate mostly to the design of management applications. In particular, management platforms lack mechanisms to automate the effortless introduction of MA-based management tasks and also to dynamically modify the functionality of distributed management services, i.e. to upgrade services at run time. Moreover, these platforms define simplistic, two-level MA organisation models, which are not suitable for the management of large-scale, geographically dispersed enterprise networks and does not allow the dynamic adaptation of the management framework to the changing topology and traffic characteristics of such networks. In this work these issues have been addressed through a design of scalable and responsive MA-based management architecture.

Referring to MA-based management applications, most of the approaches concentrate on the mobility aspect of the paradigm, overlooking the optimised itinerary aspect of the MAs. When MA follows an arbitrary route, it is very likely that it traverses on the link having a high bandwidth utilization and greater round trip time thereby increasing 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. This thesis compares the performance of client server static agent based network management to MA based network management and applications.
1.3 MOTIVATION

Modern day telecommunication industry is facing ever increasing demand for more sophisticated services, higher quality and shorter times to market. Network operators and service providers in the telecommunications industry must meet these demands at a commercially viable cost and should have plans in place for any unforeseen change. This flexibility must be achieved in an environment that is complicated by distributed, often mobile, data, resources, service access and control [11], especially when these networks are growing in size and complexity. Moreover, varied technologies, such as SONET, ATM, Ethernet, 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.

In recent years we have witnessed an evolution in network management approaches driven by the need of addressing the requirements of modern networks becoming increasingly large, sophisticated and complex. Traditionally, the Network Management (NM) scene has been dominated by simple network management protocol (SNMP) for data networks offered by the Internet Engineering Task Force (IETF) and the OSI common management information protocol (CMIP) for telecommunication network [14]. These protocol frameworks require network operators at NMS level to make real-time decisions and manually find solutions for the series of problems in the network. It is commonly accepted that the management of large-scale networking environments cannot be adequately addressed through centralised approaches that characterise contemporary management systems.

In centralized client/server (CS) approach 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 as shown in Figure 1.2.
The functionality of both managers and the agents where managers act as client and the agents as a server is statically defined at design time. Concentration of functionality, intelligence and responsibilities at a central site represents a rigid management solution that suffers from the well-known problems of centralised systems [6]. These network management systems deal only with data gathering and reporting methods, which in general involve substantial transmission of management data thereby consuming a lot of bandwidth and computational overhead. Besides this centralized management activities are limited in their capability as they cannot do intelligent processing like upfront judgment, forecasting, analyzing data and make positive efforts to maintain quality of service [7].

While a protocol-based approach is specific to a management framework, a generic approach to the client/server model can be achieved through the use of a distributed object framework as proposed by researchers in the mid 90’s. These are Common Object Request Broker Architecture (CORBA) [19] proposed by the Object Management Group (OMG) and Java-Remote Method Invocation (Java-RMI) [18] proposed by Sun Microsystems. Distributed object frameworks, allow the creation of decentralized, static
systems. Decentralization is achieved by placing required management logic in network nodes and by creating instances of management objects specific to interested clients.

Nevertheless, distributed object frameworks still suffer from a lack of support for programmability, as the management logic located in network nodes is static and cannot be easily altered. The issue of programmability of managed nodes is particularly important for network management systems. The complexity and long standardization cycles associated with TMN systems crucially hindered their success. Network managers of such systems had to wait for several years before a standardization cycle was complete and the required management functionality was embedded in network nodes.

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 [8] [9]. However, these models are typically identified by static management components that cannot adapt to the evolving nature of today’s networks, characterized by 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 [10]. One of the most popular topics in MA research community has been distributed NM, 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). Network management based-on Mobile agent refers to equipping agents with network management intelligence and allowing them to issue requests to managed devices/objects after migrating close to them as depicted in Figure 1.3.
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.

1.4 CHALLENGES IN IMPLEMENTING A SCALABLE NETWORK MANAGEMENT ARCHITECTURE

A critical look at the available literature indicates the following issues that need to be addressed towards building scalable and responsive network management architecture:
1. **Network overhead due to Polling-based nature of monitoring:** Polling-based nature of SNMP for network monitoring operations and data collection process [28] causes high network overhead. The manager repeatedly sends request and retrieves specific MIB object values at each poll cycle from remote agents. For many monitoring applications, this may cause considerable portion of network bandwidth to be wasted to learn nothing new other than that the network is operating within acceptable parametrical boundary conditions.

   **Solution:** An MA-based solution has been proposed where an agent migrates to managed devices and collect the needed information locally. This saves number of remote interactions wasting the network bandwidth. Further, MA adopts an event-driven approach wherein, managers are notified about the changes through change notification events.

2. **Centralized Management/ Single Point of Failure:** In centralized Client/Server model [59, 60, 69, 70], network manager 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.

   **Solution:** Proposed architecture allows management intelligence to be embedded in the mobile agents. As a result management components and services are distributed in nature and they move from host to host irrespective of underneath management model. Mobile agents can survive intermittent network failures and can easily work off-line and communicate their results when the application is back on-line.

3. **Scalability:** In centralized Client/Server model, all the management data is transported to management station for management decision. This doesn’t scale as the network grows in complexity and size [6, 7, 80].
Solution: Proposed architectures prevent the overloading of the network by dispatching mobile agents to remote devices where mobile agents process the data locally rather than transferring over the network. By deploying mobile agents the management data around the management station reduces significantly.

4. Processing bottleneck at Management Station: The centralised nature of Client/Server architecture and the lack of data filtering capability of static agents residing on managed devices, result in transferring of large amount of network management data to the manager platform for processing. This forces manager’s processing capacity to its limits and intensifies the need to relieve the manager from performing routine data processing tasks [70, 80].

Solution: In MA-based proposed solution the computational load at the MS is reduced nearly to zero as the management tasks are executed locally on the NEs involved in the task.

5. Bandwidth and Latency issues: 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. Moreover, 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.

Solution: In MA-based proposed solution Mobile agents offer an alternative because they can be dispatched from a central controller to act locally and execute the controller’s directions directly. This will save on time & network bandwidth wastage from large amount of network management traffic.
6. **Flat-Bed Model:** Most of the MA-based frameworks proposed for network monitoring applications assume a ‘flat’ network structure [70], i.e. a single MA is launched from the manager platform and sequentially visits all the managed NEs, regardless from the underlying topology. This only partially solves the scalability limitations of centralized architectures.

**Solution:** The proposed IMASNM & EM based Models offers domain partitioned hierarchical management architectures where mobile mid level managers manage their own domains and keep all the management data confined to their domains only. Global managers are notified about local management reports from sub-domain managers. This saves on expensive travel of a single MA through domain interconnecting costly links (in terms of bandwidth).

7. **Static Middle Level Managers:** Use of static mid-level managers [66, 70], in between the management applications and managed devices, for distributed, hierarchical, MA-based models pose flexibility and scalability issues.

**Solution:** The proposed MA-based solution introduces a highly scalable, flexible hierarchical management model with the ability to dynamically adapt to the evolving network conditions both in terms of topological changes and traffic characteristics of large-scale enterprise networks.

8. **Mobile Agency Agency:** Models talk about deploying MA as mid-level managers on network devices for managing their own domains or zones. Many of these devices don’t support mobile agent runtime environment. Little attention has been paid on how to manage these devices without mobile agent execution environment like hubs and switches in enterprise networking environment and most of the telecomm devices like terminals, switches and multiplexers. This also includes legacy devices [81].
Solution: Separate EM based platform is proposed to run Mobile Agent Agencies/Environment so as to relieve managed devices from hosting these agencies and increasing the footprint of limited memory.

9. Static Service Models: Most of MA-based models provide Static Service Models where it is not easy for the administrators to easily add new services or modify existing ones at runtime with minimum disruption to application [82, 83].

Solution: To facilitate dynamic service reconfiguration including introduction of new service, upgrading of existing services and unloading of services no longer needed a Dynamic Service Manager Module is proposed.

1.5 ORGANIZATION OF THESIS

The thesis is organized as follows:

Chapter 2: The basic concepts of network management systems and the architecture of a general network management application have been briefly discussed in this chapter. A detailed literature review of available standards, protocol framework and existing architectures in the field of network management systems has been provided. The review also considers the role of distributed techniques and application of mobile agents in the field of network management.

Chapter 3: In this chapter, a general analytical model and framework for the evaluation of various network management paradigms is introduced. It is also illustrated as to how the developed analytical framework can be used to quantitatively evaluate the performances and tradeoffs in the various computing paradigms. The performances of the mobile agent based paradigm with the corresponding ones under the Client-Server mode under different scenarios have been compared.

Chapter 4: In this chapter the management of hierarchical, geographically dispersed networks. In particular, it introduces a highly scalable and flexible MA-based hierarchical
management model with the ability to dynamically adapt to the evolving topological and traffic characteristics of large-scale enterprise networks. The scalability of the proposed framework is, again, verified by a quantitative evaluation and empirical results.

**Chapter 5:** This chapter presents a domain partitioned network management model based-on mobile agent & Element Management Systems in order to minimize management data flow to a centralized server. Intelligent agent allocated to specific EMS performs local network management and reports the results to the superior manager and finally the global manager performs global network management using those submitted management results. Concepts of dynamic service management & minimum spanning tree based itinerary design are also discussed.

**Chapter 6:** In this chapter optical transport network is being introduced as a kind of WDM optical network. The setting up of an optical path within the OTN is a challenge in the presence of heterogeneous devices connected on the network and availability of optical ports on them. The cost of setting up the optical path (also called \( \lambda \) - connections) for both existing (C/S) and proposed (MA-based) models has been computed with a view to compare their performances.

**Chapter 7:** It concludes the outcome of the work proposed in this thesis. It also endeavors to explore the possibilities of future research work based on the proposed design.