

# MODELING AND DESIGNING RESOURCE EFFICIENT DISTRIBUTED MANET MANAGEMENT SYSTEM (DMMS)

Golam R. Khan<sup>1</sup> and Dhadesugoor R. Vaman<sup>2</sup>

Department of Electrical and Computer Engineering, Prairie View A&M University,  
Prairie View, Texas 77446, USA

<sup>1</sup>– Doctoral Candidate;

<sup>2</sup>–Texas A&M University System Regents Professor;

## **ABSTRACT**

*Traditional network management systems use spoofing of resources to collect statistical data such as resource usage and performance. This data is exchanged with other resources through management protocols. The amount of data can be extremely high and the bandwidth for overhead management functions increases significantly. Also, the data storage requirements in each network resource for management functions increases and become inefficient as it increases the power usage for processing. In this paper, we propose a distributed network management system where each network resource maintains a set of Management Information Base (MIB) elements and stores resource activities in their abstraction in terms of counters, flag and threshold values. The abstract data is exchanged between different management agents residing in different resources on a need-to-know basis and each agent logically executes management functions locally to develop understanding of the behavior of all network resources to ensure that user protocols can function smoothly. Our proposed architecture can be implemented in any network, but is highly important to be considered in power and bandwidth constrained networks such as Mobile Ad Hoc Networks (MANET). In this research, we use cross layer models to demonstrate simplified way of efficiently managing the overall performance of individual network resources (nodes) and the network itself which is critical for not only monitoring the traffic, but also dynamically controlling the end-to-end Quality of Service (QoS) for multi-service applications.*

## **KEYWORDS**

*Network management, MANET, Distributed MANET Management, MIB, Managed Objects (MO)*

## **1. INTRODUCTION**

Network management is defined as a process for monitoring and controlling real-time network behavior. This will allow provisioning multiservice applications with desired Quality of Service (QoS) accurate end-to-end. [1]. This process involves management agents in different network nodes to remotely access the management attributes stored in each local Management Information Base (MIB) associated with nodes in order to facilitate network monitoring and real time remote control of behavior of user protocol layers. The attributes are coded in terms of counter values, flags and thresholds, which represents the user protocol operation at each local node. The International Standards Organization (ISO) defines OSI management model which specifically defines different management functions [1-2]:

- Fault management,
- Accounting management,
- Configuration management,
- Performance management, and
- Security management.

At any instance of time, the operational control of the network resources are in the hands of these management functions. When the network is operating in normal mode, the control is in the hands of performance management. When any resource experiences a fault, the fault management controls the resource. When a new resource is being configured, the configuration management takes control; when any resource has a security violation, the security management takes control and the accounting management controls the resource cost and usage management. These functional management components are designed modularly and each of these functions looks at the MIB elements to take any actions whether it is real time or non-real time. Thus, there is some abstraction is maintained to simplify the management functionality of the network.

The MIB elements which depict the behavior of the network resources are defined in terms of Managed Objects (MOs) and associated attributes. The attributes are typically specified in terms counters, thresholds and flags.

In this paper, we show a design of distributed network management for monitoring and controlling the behavior of applications supported for each source-destination pair of nodes in Mobile Ad Hoc Networks (MANET).

MANET is an ad hoc network which does not use towers or base stations and can be defined as a system of autonomous mobile (Dynamic) nodes that communicate over wireless links without any preinstalled infrastructure. The network is both power and bandwidth constrained and yet it is expected to provide multi-service provisioning with end-to-end QoS provisioning to end-users.

It is significantly harder to manage MANET for the following reasons:

- The topology of a MANET is quite dynamic due to high mobility of nodes and the connectivity between a source and a destination is maintaining using multi-hop node connectivity.
- The resources (nodes and sensors) have limited battery power, bandwidth constrained links with variability. As a result, management overhead has to be limited to allow the bandwidth to be used for user applications maximally.

The design of network management architectures uses either centralized (a single network manager) or hierarchical (maintaining multi-layered network manager) or distributed (allow each node to have an agent which takes manager-ship when initiating an action or responds to a command).

Centralized management where only one manager exists for the network does not allow scalability of the network as too much bandwidth can be consumed for exchange of management information. Hierarchical management system can be designed for MANET, but it is not efficient in using the bandwidth and requires reconfiguration of the network due to high mobility of all nodes. Thus, it is important to use distributed network management where a management agent in a node takes manager-ship for a small duration of time and relinquished once the execution of an activity is done. This allows every node to be acting as manager at different instances of time. This will allow designing scalable networks with any number of nodes.

In MANET, several nodes that are within close proximity of one another at an instance of time are grouped into a cluster and each cluster has a designated Cluster Head (CH) node to act as a manager. Also, when a node communicates from a cluster to another node outside the cluster, the node needs to access the remote node through CH. Thus, CH tends to be vehicle mounted where the power is not an issue.

In this paper, we present a design of a distributed management architecture which supports both performance monitoring and real time control of performance for multi-service applications with minimal use of bandwidth in the network for management functions.

The remaining sections are organized as follows:

- Section 2 Background Research
- Section 3 Management Systems Design – Distributed MANET Management Architecture (DMMS)
- Section 4 Operational Scenario of DMMS
- Section 5 DMMS Support of Applications
- Section 6 Conclusions

## **2. BACKGROUND RESEARCH**

Traditional management systems for voice and data networks were designed for Operations, Administration and Monitoring (OA&M). They did not have the real-time controlling functional. We have also seen that traditional Network Management methods are developed to manage wired networks. The characteristics of traditional network management are quite different from the characteristics that are used in MANET. There are some researches have done for network management of ad-hoc network [3][4][5] and Wireless Sensor Networks (WSNs) [6][7][8] those are based on Simple Network Management Protocol (SNMP) where polling mechanism is used. Since MANET can be consists with a large number of nodes and multi hop mechanism is used for delivering the data through the network, polling management messages in MANETs sometimes causes huge amount of management traffic. Again, the performance of the network will be affected as management messages uses resources of MANETs. Therefore, polling is not appropriate to MANET, which need to be resource efficient as well as power efficient. In this research, we proposed the management message exchange method and two types of operational scenarios for management purpose to reducing the polling operation. ANMP [3] uses centralize approach where the management system requires monitoring and controlling all the network resources. It also allows hierarchical clustering of management nodes to reduce the number of messages exchanged between the manager and the agents (mobile nodes). Maintaining hierarchical system of management nodes is cost ineffective and adverse to handling mobility of nodes rapidly as exchanges between management nodes about node mobility is slower and network configurations are often faster. As such, normal data operations can suffer from errors. In addition, the requirement to prevent single point failures is expensive as it is has to duplicate nodes significantly.

In ANPM [9], they presented the proposed autonomic network performance management architecture and modules. Their case study focused on centralized implementation. However they did not mansion about the fully distributed or hybrid approach. They showed the different network performance such as throughput and end-to-end delay etc. In our proposed DMMS, we have shown a simplified way to find the network performance by using cross layer management objects (MOs) and distributed the management functions among all the cluster members.

In WSNMS [10], they designed and implemented with the characteristics of WSNs. They showed an actual experiment that consists of large number of nodes with managed objects and a gateway. In that work, WSNMS is able to manage the large-scale network management functions and provide good performance. But WSNMS has some drawbacks, such as it has only a manager, which causes high communicate overhead from data polling and single point failure. In our proposed system, we have used cluster based distributed network management where each cluster has a cluster head or a manager that able to reduce the high message overhead.

### 3. MANAGEMENT SYSTEM DESIGN

#### 3.1. Proposed Distributed MANET Management Architecture

Figure 1, we shows the architectural representation of the proposed Distributed MANET Management Architecture (DMMA). In this architecture, each node has an agent with an associated distributed MANET management system MIB (dMmsMIB). The network management operation is based upon one of the agents to assume managership (become manager) and communicate with a remote agent for accessing the dMmsMIB elements for either monitoring of the network behavior or for controlling the network behavior in real time. Any agent can assume managership at an instance of time. Figure 1 also illustrates the MANET in terms of clusters and communications across the multi-clusters require intra-cluster communications as well as inter-cluster communications.

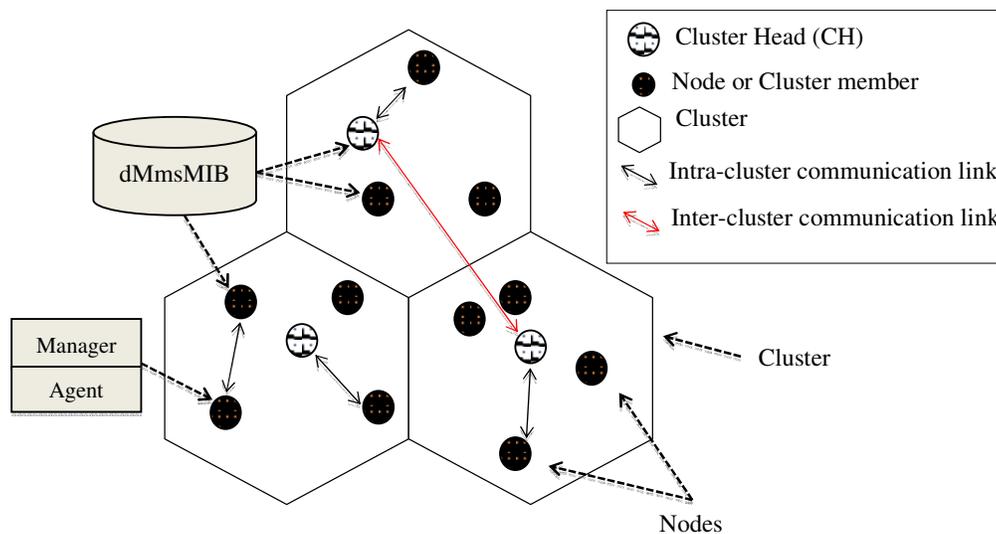


Figure 1. Architecture of the Distributed MANET Management System (DMMS)

Each cluster is formed with a set of nodes and an associated cluster head (CH). Management protocol for both intra-cluster and inter-cluster communications will be using defacto standard, Simple Network Management Protocol (SNMP). For the paper, SNMP is irrelevant. CHs have the responsibility of managing their own clusters based on a pre-set of policies. From a management perspective, each CH can either send MIB information to a selected node or broadcast to all its associated nodes depending on the nature of management operation. It can send individual managed object with a set of attributes or send a collection of elements with attributes.

### 3.2. Management Message Exchange

#### 3.2.1. Message Types

In our proposed manager-agent communication, we have divided communications into three parts. First one is to get the attribute from a remote agent/s, second one is to set the management attributes in the MIB operated by a remote agent/s and the last one is to allow sending of a trap message by an agent to handle abnormalities.

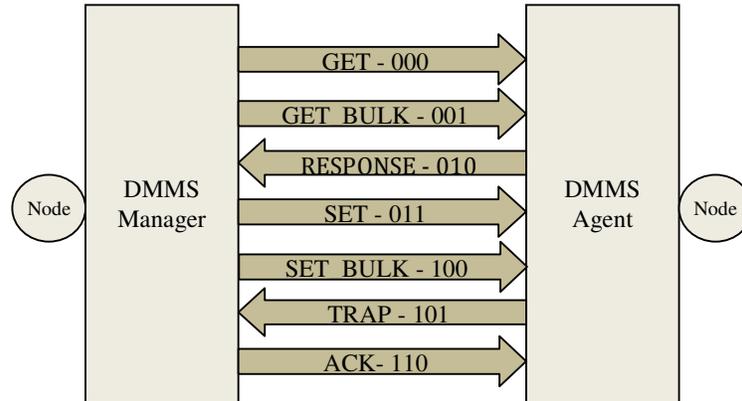


Figure 2. Manager-agent request, response and trap message flow

We also proposed the usage of management messages. In Figure 2, we show 7 (seven) message types. GET, GET\_BULK, SET, SET\_BULK, RESPONSE, TRAP and ACK, which are quite similar to SNMP. But the usage is different from SNMP. In DMMS, since the network status such as channel condition, battery power depletions, topology changes and faults changes dynamically, it is necessary to get the information immediately and for getting those information we defined those messages types. Since in MANET environment, nodes are scarce and mobile, polling for data by an agent increases delay and adds overhead traffic. However, by using MIB exchanges in abstract form, it is possible to reduce the overhead use of bandwidth. In this paper, we proposed an efficient MIB exchanges to design efficient management system.

#### 3.2.2. Manager and Agent Communication Process

We assume the communication process between manager and agent in our proposed DMMS as follows:

- Each agent of any node can act as a manager to communicate with other nodes.
- In DMMS, the manager can send request to an agent to operate on a set of MIB elements at the remote location. Example of operation includes setting a MIB element or getting a MIB element.
- For each request, a remote agent must send a *RESPONSE* and/or a *TRAP* message to the manager.
- When the manager sends *GET* type PDU to an agent, it sets the request id, node id, Managed Object ID and all IDs of associated attributes (we define unique attribute id for each attribute). This PDU is sent to a remote agent of a remote node. The Agent processes the GET request by sending an acknowledgement for the requested information to the manager, which includes the contents of all of the requested attributes. This is done in terms of a RESPONSE PDU type.

- When using a *GET\_BULK* type PDU, the manager broadcasts a PDU to all the nodes of the cluster to send the necessary attributes from each of the MIBs of remote agents. This is typically used when common attributes are requested. Each Agent node sends a *RESPONSE* to the manager with the contents of the requested attributes. *RESPONSE* PDU contain only the content of the MIB attribute from the associated node only. If node id of that requested PDU correspond to the agent's node id, the request is intended for the associated remote agent. Then only, the remote agent is allowed to send a *RESPONSE* PDU with the necessary contents of the MIB attributes. All *RESPONSE* PDU should contain the same request id and their own node id.
- *SET* type PDU is used to set the management attributes in the MIB of a remote agent. The sending Manager enters the attributes' ids and their respective values in the PDU which is sent to the remote agent. Upon completion of "set" process, the remote agent sends an ACK PDU to the manager. This type of setting process is used for configuration management during initialization of a new software or a system or for a fault manager to enter the MIB elements to detect the location of the fault and clear the fault remotely.

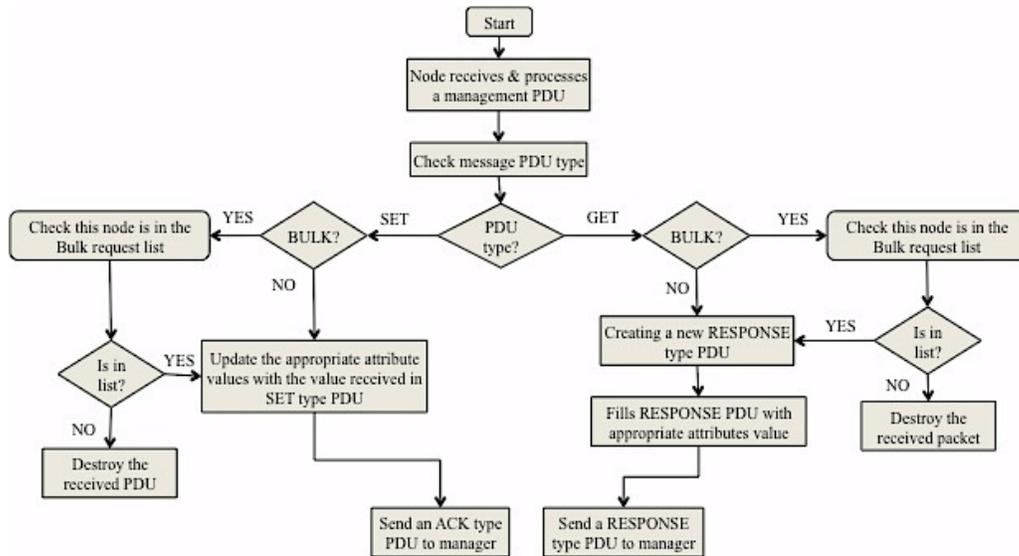


Figure 3. Activity diagram of an agent node after receiving a management message

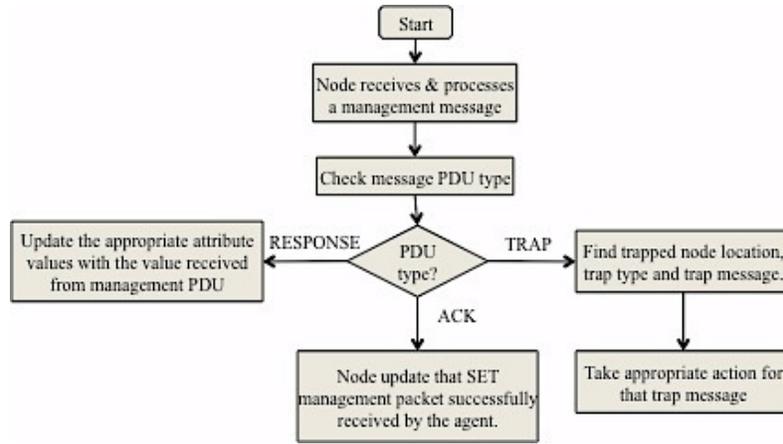


Figure 4. Activity diagram of a manager node after receiving a management message

- Manager broadcast the *SET\_BULK* type PDU to the cluster members. In this PDU, The sending Manager enters the attributes' ids and their respective values in the PDU and broadcast it. All the agents of associated nodes within the cluster processes the SET operation. This operation is critical for sending snap shots such as PL&T locations of all the nodes in the cluster to each and every node within the cluster. Each node can make use of the available information for normal data operations such as setting up the routes or provide uninterrupted service provisioning using the knowledge of the node locations.
- *TRAP* message is used for communicating from an agent to a manager for informing that an exceptional event has occurred and need immediate attention. This is specifically useful for fault management.

In Figure 3 and 4, we show the flow diagram of an agent and of a manager after receiving a management packet where an agent can receive GET or GET\_BULK or SET or SET\_BULK request and a manager can receive RESPONSE or ACK or TRAP message.

### 3.2.3. Management Information Bases (MIBs)

Figure 5 shows some example of MIBs that we proposed to use in our distributed MANET management system's MIB (dMmsMIB). We define two type of management PDU for DMMS. First one is for sending GET, SET and RESPONSE operation and second one is for sending trap message those are shown in Figure 6. In MANET environment, some information is changed continuously (e.g. number of packet send, number of packet receive, and number of packet loss in a particular threshold) and some information is not (e.g. node id, device manual id, and coverage info). Both types of MIBs need to be handled differently. We defined two types of management operations such as normal basis management and need-to-know basis management. In normal basis management operation, dynamic MIBs are exchanged. In need-to-know basis management operation, both dynamic and static information are exchanged.

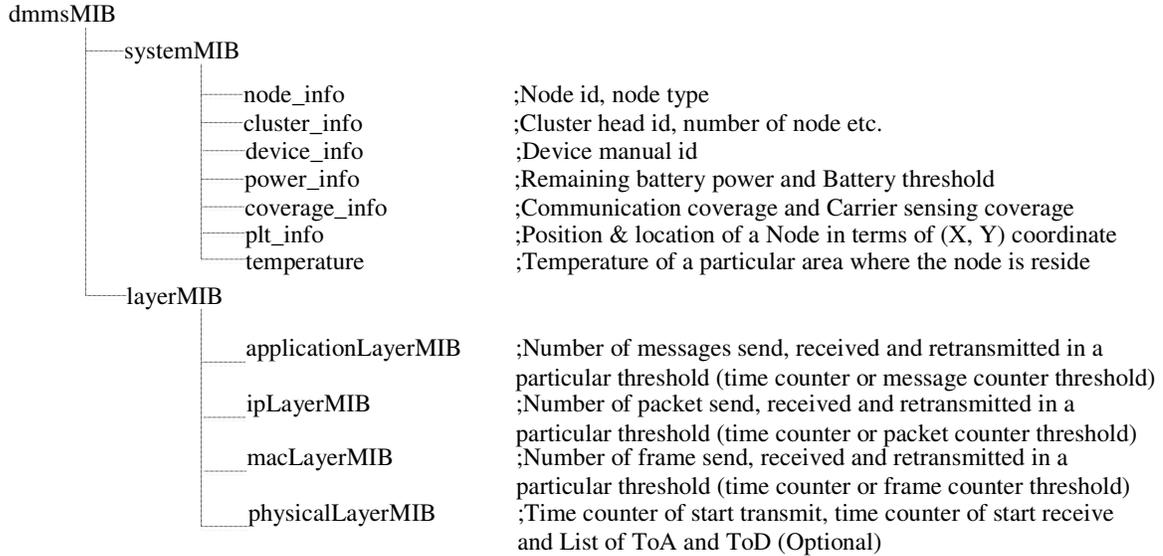


Figure 5. Distributed MANET management system's MIBs (dmmsMIBs)

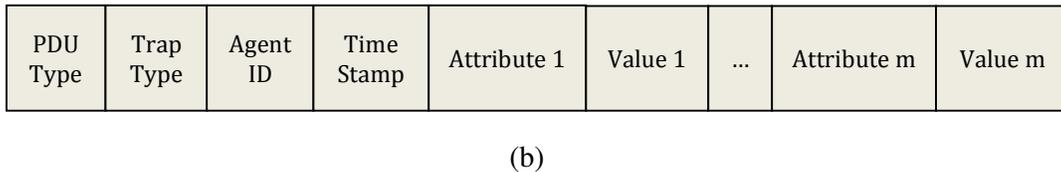
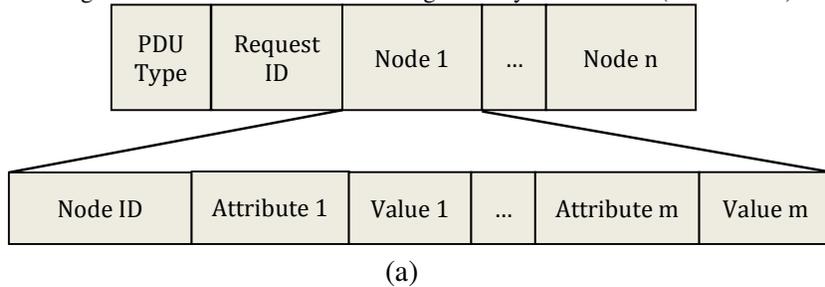


Figure 6. Two different types of PDU for (a) get, set, response and (b) trap request

## 4. OPERATION SCENARIO OF DMMS

### 4.1. Normal management operation

All the nodes in a cluster normally send and receive packets from other nodes. During any transmission, when a node observes that an attribute (e.g. number of packet count) has reached to a threshold, it will send a broadcast or unicast message (management) like “*SET\_BULK PDU*” or “*SET PDU*” or “*TRAP PDU*” to another node. We define this type of management operation as normal management operation. Trap message is a normal management operation because when any node observes an abnormal operation such as too many error packets in a row, remaining battery life reached the threshold, node is dying and communication link quality becomes bad, that node sends TRAP messages to the sender node or to the CH depending on which type of trap message it is. In section 4, we will simulate some regular management operations and implement some management applications (e.g. throughput, delay, offered data rate). Figure 7 shows the flow diagram of recording management information.

## 4.2. Need to Know Basis Management

Sending management information periodically to the cluster members is cost ineffective that implies too many management packets causes high message overhead and bandwidth wastes. In regular basis management, we sends only one management packet in each ensemble, which cost low message, overhead. Need-to-know basis management allows us the efficient use of the resources. In this management, when manager wants the information, they can request the agent to send specific management information those are needed to know. Normally cluster head (CH) to cluster member nodes or one cluster's CH to another cluster's CH are communicating using need to know basis management operation. Furthermore, the message overhead between node-to-node communications cost low message overhead because nodes are getting the attributes according to their need. *GET* and *GET\_BULK* are two PDU type packets that normally send to the agent nodes when any manager needs to know some management attributes. Agent node gives the reply by sending the *Response* PDU type packet. Sometimes Cluster Head (CH) needs to know the MIBs from other cluster member nodes and on that time CH broadcast the *GET\_BULK* type PDU.

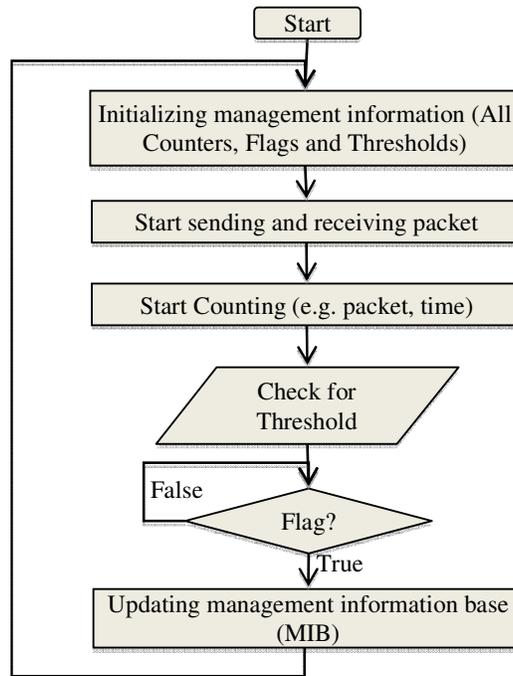


Figure 7. Recording management information

## 5. DMMS SUPPORT OF APPLICATIONS

The use of distributed network management with MIB elements for control and monitoring is extremely cost effective compared to conventional network management methods where expensive devices are used to monitor the network at different locations of the network globally. In our proposed distributed network management approach, each source-destination pair controls their applications independently without requiring any knowledge to be accessed from the network itself. In this section, we have shown some management applications that can be used for controlling and monitoring purpose.

## 5.1. Controlling Purpose

We can use our DMMS for controlling MANET routing. In MANET environment nodes move randomly. Therefore, to know the position of the nodes at any time is really important for successful routing. In cluster base MANET; normally cluster head (CH) have many responsibilities to manage the cluster and to maintain the communication between two clusters. In our proposed Distributed Network Management (DNM), we can distribute management functions to cluster members or nodes to reduce the load of CH. By analyzing one of the graph theoretical routing protocols [11], we have discussed how DMMS can use in controlling purpose to manage CH's load.

In [11], cluster head uses the Position Location and Tracking (PL&T) system [12] of all the nodes within the cluster and develops a snap shot of the cluster. The CH periodically broadcast the snap shots to ensure nodes know as and when PL&Ts of nodes have changed. If any routing is on going and some of the node's location has changed but the updated snapshots were not received from the CH, on that time routing can be failed. In Graph Theoretic Routing [11], receiving the updated or correct snapshots is very important for efficient routing. Management part acts as a big role for controlling the PL&T locations snapshots. When any routing initialized, nodes find the degree by using the snapshot that send by the CH. Management agent double check the PL&T location of those nodes which will use to take place the routing. By checking the location again, routing can be successful and accurate.

## 5.2. Monitoring Purpose

In our proposed DMMS, we have shown some simplified way to find different network performance such as throughput, data rate, delay and etc. for monitoring purpose. To calculate those performances, we have defined some layer management objects (MOs) for each layer in terms of abstraction form of data. Each node has the same layer MOs to perform those network performance and any node can get information of other nodes by using need to know basis operation. In this section we have shown some management applications that can be used for monitoring purpose. In Table 1, we have given some list of QoS applications can be used by layer MOs.

Table 1. List of QoS applications used by layer MOs

Layer	Attributes
<b>Application</b>	<ul style="list-style-type: none"> <li>• Information Rate Target</li> <li>• Loss of Information Target</li> <li>• Response Time Target</li> </ul>
<b>IP</b>	<ul style="list-style-type: none"> <li>• Packet Delivery Rate</li> <li>• Packet Loss</li> <li>• End-to-End Delay</li> </ul>
<b>MAC</b>	<ul style="list-style-type: none"> <li>• Frame Delivery Rate</li> <li>• Undetected Frame Error Rate</li> <li>• Average and Maximum Frame Delay</li> </ul>

### 5.2.1. Throughput and Offered Data Rate

To compute throughput, offered message rate, offered packet rate, offered frame rate, offered data rate, maximum data rate and average data rate, we define Application layer MO, MAC layer MO and IP layer MO in terms of counters, thresholds and flags. When nodes start sending a new

ensemble of data, nodes initialize their time counter, number of packets send counter, number of packets received counter and number of packets retransmit counter. In Figure 7, we have shown the update process of MIB or management attributes of each layer MOs. When nodes start sending every ensemble of packets, first packet is a management packet. In this management packet contain the management attributes or MIBs those are used to compute the different network performance. Application layer MO, MAC layer MO and IP layer MO are used to find the message, frame and packet related network performance respectively.

For simplicity we have just shown IP layer MO to find some network behaviors of packets. By using IP layer MO, node can find the offered packet rate, packet throughput, maximum data rate and average data rate. In Figure 8, IP layer MO is shown with its management attributes and description of each attributes are also shown in comments. *Packet\_Timer\_Count* start counting the time when nodes start sending or receiving a new ensemble of packet and stop counting when *Packet\_Count* reached to the *Packet\_Count\_Threshold* limit. When the counter reached to a threshold on that time the *Flag* updated the value to false. When nodes again start sending and receiving the packet, all the counter and flag must reset to its initial value. *Max\_Packet\_Size* and *Average\_Data\_Size* are updated after every ensemble of packet sends or receive.

IP\_Layer\_MO

Packet_Timer_Count	;	Value specifies the count of time (reset initially and end value specifies the amount of time in seconds)
Packet_Count	;	Value specifies the number of packet transmitted
Packet_Repeat_Counter	;	Value specifies the number of packet repeated
Packet_Count_Threshold	;	packet count threshold or each ensemble size
Packet_Start_Flag	;	flag = true when start transmitting and flag = false when stop transmitting
Max_Packet_Size	;	Updated Max_Packet_Size when Max_Packet_Size < Packet_Size
Average_Data_Size	;	Average all Packet_Size

Figure 8. IP Layer Management Object (MO)

We have shown some equations from (1)-(5) where how to find the offered packet rate, packet throughput, maximum data rate and average data rate are shown.

$$\text{Offered Packet Rate} = \frac{\text{Packet\_Count}}{\text{Packet\_Timer\_Count}} \text{ packets/sec} \dots \dots \dots (1)$$

$$\text{Packet Throughput} = \frac{(\text{Packet\_Count} - \text{Packet\_Repeat\_Counter})}{\text{Packet\_Timer\_Count}} \text{ packets/sec} \dots \dots \dots (2)$$

$$\text{Maximum Data Rate} = \frac{(\text{Packet\_Count} * \text{Max\_Packet\_Size})}{\text{Packet\_Timer\_Count}} \text{ bits/sec} \dots \dots \dots (3)$$

$$\text{Average Data Rate} = \frac{(\text{Packet\_Count} * \text{Average\_Packet\_Size})}{\text{Packet\_Timer\_Count}} \text{ bits/sec} \dots \dots \dots (4)$$

We simulated the proposed distributed network management system using the Network Simulator 2 (NS2) for MANET. In this simulation, a single cluster of MANET is created with 10 associated nodes. The nodes in the cluster are placed randomly in a 100m x 100 m geographic area and they are allowed to move randomly as specified in Table 2.

Table 2: Simulation parameters

Parameter	Values
Area	100m x 100m
Number of Nodes	10
Node Placement Strategy	Random
Mobility	Random Min speed: 1.00 m/s Max speed: 5.00 m/s Avg. speed: 2.49 m/s

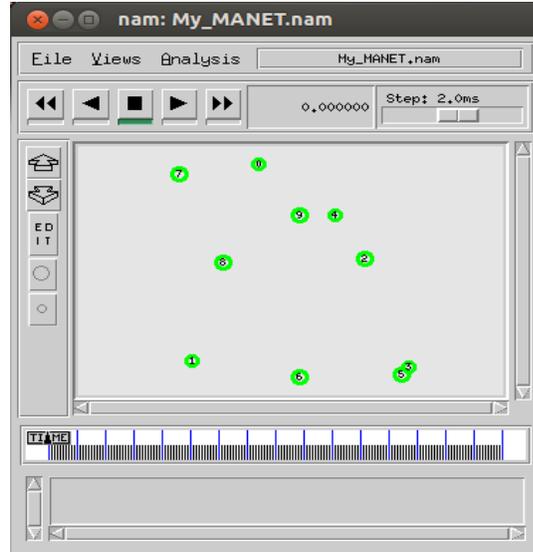


Figure 9. A cluster with the random distribution of nodes

Figure 9 shows a cluster of nodes based on the simulation parameters of Table 2. Assuming that node 0 is transmitting an ensemble of data packets to node 1, the management agent of node 0 sends a management packet which will be the first packet in each of the ensemble of data packets.

It is possible for each node to compute the number of packets sent and the total time (using a counter) required for sending all the packets will provide the total traffic offered traffic as a count within the ensemble time. Similarly at the destination node, the total number of packets received within the time count can be counted. The ratio of the number of packets transmitted from the source and the number of packets received at the destination within the ensemble time will give the throughput, which can be measured independently by both sending node and the receiving node. In Figure 10 we show the graph for throughput with respect to number of packet send. For throughput we consider the following equation and each ensemble size is 100 packets.

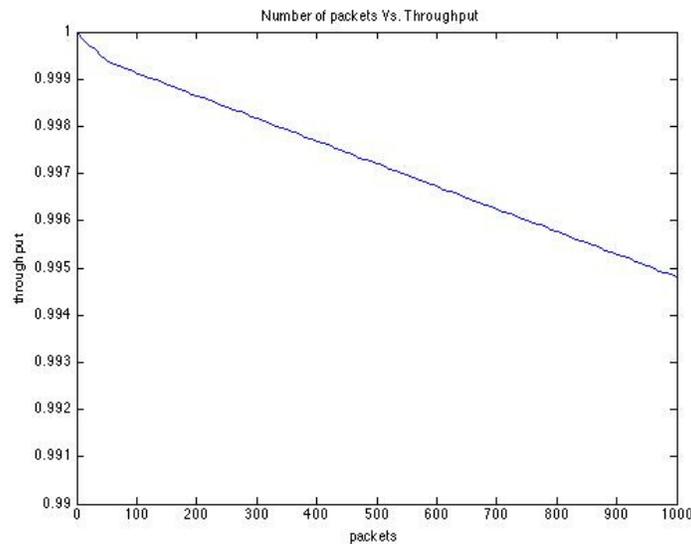


Figure 10. Throughput calculations by sending IP\_MO

By setting up the mac layer MO (MAC\_MO) and application layer MO (APP\_MO), nodes can also find out offered frame rate, frame throughput, offered message rate and message throughput.

### 5.2.2. Packet Delay

To measure the end-to-end packet delay for transfer of IP packets between the *node A* and the *node B*, how many packets are transmitted from the sending node and compute how much time it takes to receive all these packets at the receiving node. We have sent N packets and measured the time it took for sending N packets. On the other hand, we have also received N packets and measured the time it took to receive N packets. For measuring the total time it took for sending N packets, we have kept records the time of departure for the 1<sup>st</sup> and n<sup>th</sup> packet of every ensemble in sending side. Again, for measuring the total time it took for receiving N packets, we have kept records the time of arrival for the 1<sup>st</sup> and n<sup>th</sup> packet of every ensemble in receiving end. In the receiving end, we have calculated the delay by using the sending side information that is sent through the management packet after each ensemble. The total delay is the difference between the amount of time it took to receive the packets at the receiving side and the amount of time it took to send the packets at the sending side. This method to find the end-to-end delay called *long-term average method*.

#### PHY\_Layer\_MO

Transmit_Start_Time_Counter (A)	; Counter value specifies the time. Nodes start counting time when packets sending begin
Receive_Start_Time_Counter (B)	; ; Counter value specifies the time. Nodes start counting time when packets receiving begin
Transmit_Time_Counter_Threshold (N)	; Specifies the maximum number of packets in the transmit ensemble (integer) = N
Receive_Time_Counter_Threshold (N)	; Specifies the maximum number of packets in the received ensemble (integer) = N
Transmit_Timer_Threshold_Flag	; Boolean is initially set to "0", 0 remains when the A less than N; when A equals N, Boolean = 1
Receive_Timer_Threshold_Flag	; Boolean is initially set to "0", 0 remains when the A less than N; when A equals N, Boolean = 1

Figure 11. Physical Layer Management Object (MO)

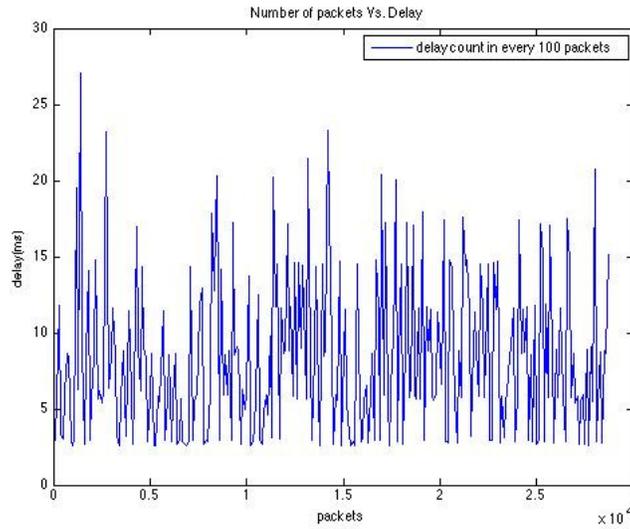


Figure 12. Delay of each ensemble of packet by using long-term average method

We have also simulated a cluster of 10 nodes to show end-to-end delay. Both IP layer MO and physical layer MO are used to find the end-to-end delay. In the proposed management approach for computing the time delay, we have established the following attributes at the physical layer as part of *PHY\_Layer\_MO* in Figure 11 and IP layer MO have shown already in Figure 8.

Figure 12 shows the graph of the delay for every ensemble of packets where each ensemble consists of 100 packets. Figure 13 shows the graph of average delay that is calculated after each ensemble of packets.

Through above discussion, we can say that by setting some attributes in each layer MO, we can control and monitor different applications. In traditional network management, to control and monitor those applications, they use some devices to find those attributes. However the message overheads and resource usages in each node are too high. In our proposed DMMS, the message overheads and resource usages are less than the traditional management system. Firstly, there is no additional device required to find those attributes. Secondly, management task is distributed to each node find their own attributes and measure the performance matrix. Thirdly, more nodes can accommodate in each cluster because workload of the cluster head reduces by distributing the management task. Finally, as each node gets the information about other nodes based on need to know basic, message overhead reduced significantly.

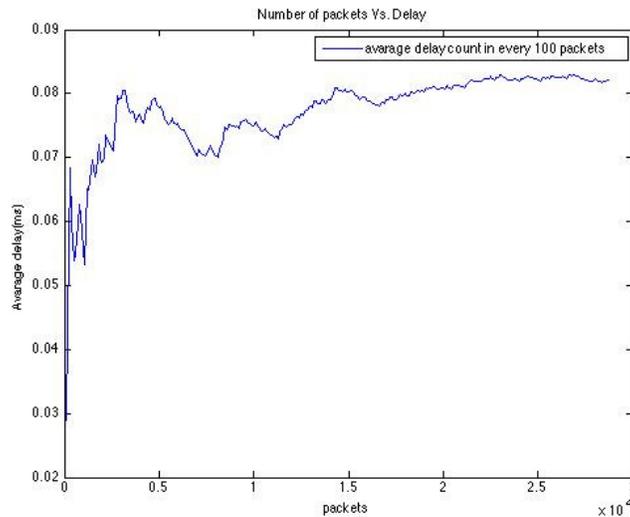


Figure 13. Average delays of packet by using long-term average method

## 6. CONCLUSIONS

In this paper, we introduced the Distributed MANET Management System (DMMS), which uses fully distributed management architecture. We have shown that the DMMS gets rid of the general manager and the “single point of failure” problems. We developed a new set of Management Information Base (MIB) elements and stored resource activities in terms of counters, flag and threshold values. We have shown that the abstract data is exchanged between different management agents residing in different resources on a need-to-know basis and each agent logically executes management functions locally to develop understanding of the behavior of all network resources for monitoring purposes (finding throughput, delay, packet loss rate) and controlling purposes (managing routing algorithm). In this dissertation, we presented the MANET architecture with MIB based distributed network management and demonstrated power and bandwidth efficiency for management operations where management operations includes monitoring of resources, and performance management for multi-service provisioning.

## ACKNOWLEDGEMENTS

This work is supported in part by the US Army Research Office under the Research Cooperative Agreement Grant, W911NF-04-2-0054 to the ARO Center for Battlefield Communications (CeBCom), National Science Foundation under the contract 0931679, Department of Electrical and Computer Engineering, Prairie View A&M University (Texas A&M University System). The views and conclusions contained in this dissertation are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government.

## REFERENCES

- [1] Leinwand, A. & Fang, K., (1995) "Network Management: A Practical Perspective", 2<sup>nd</sup> edition, Addison-Wesley publisher.
- [2] Stallings, W. (1993) "SNMP, SNMPv2, and CMIP: The Practical Guide to Network Management Standards", 1<sup>st</sup> edition, Addison-Wesley publisher.
- [3] Chen, W., Jain, N. & Singh, S. (1999) "ANMP: Ad Hoc Network Management Protocol", IEEE Journal on Select Areas Communication, Vol 1, No. 8, pp. 1506-1531.
- [4] Shen, C., Jaikao, C., Srisathapornphat, C. & Huang Z. (2002) "The Guerrilla Management Architecture for Ad hoc Networks" MILCOM 2002. Proceedings (Volume: 1).
- [5] Ayari, M., Kamoun, F. & Pujolle, G. (2008) "Towards Autonomous Mobile Ad Hoc Networks: A Distributed Policy-Based Management Approach", Wireless and Mobile Communications, ICWMC 08, Page(s): 201 – 206.
- [6] Ruiz, L.B., Nogueira, J.M. & Loureiro, A.A.F. (2003), "MANNA: A Management Architecture for Wireless Sensor Networks", IEEE communications Magazine 41(2), 116–125.
- [7] Kim, J. & Jeon, H & Lee, J (2010), "Network Management Framework for Wireless Sensor Networks (WSN)," Held as Part of the Future Generation Information Technology Conference (FGIT), Proceedings, Part I.
- [8] Budhaditya, D., Bhatnager, S., Nath, B. (2001) "A Topology Discovery Algorithm for Sensor Network with Application to Network Management", Technical Report DCS-TR-441, Department of Computer Science, Rutgers University.
- [9] Moursy, A., Xu, B., Perkins, D. & Bayoumi, M. (2010), "Towards Autonomic Network Performance Management in Mobile Ad Hoc Networks," IEEE GLOBECOM Workshops (GC Wkshps), Page(s): 448 - 453.
- [10] Zhao, Z., Huangfu, W., Liu, Y. & Sun, L. (2011) "Design and Implementation of Network Management System for Large-Scale Wireless Sensor Networks", Seventh International Conference on Mobile Ad-hoc and Sensor Networks (MSN), Page(s): 130-137.
- [11] K. Patel, D.R.Vaman (2013), "Graph Theoretic Routing Algorithm (GTRA) For Mobile Ad-Hoc Networks (MANET)", International Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks (GRAPH-HOC), Vol 5, No. 4.
- [12] N. Shakhakarmi, D. R. Vaman (2012) "Dynamic PL&T using Two Reference Nodes Equipped with Steered Directional Antenna for Significant PL&T Accuracy", Wireless Telecommunications Symposium, London, UK.

## Authors

**Mr. Golam Khan** is currently Ph.D candidate in Electrical Engineering from Prairie View A&M University (PVAMU), TX and working as a Graduate Teaching Assistant in the same Department. Previously, he was working as a Graduate Research Assistant in the Center of Battlefield Communication (CeBCom). His advisor is Prof. Dhadesugoor R. Vaman. His current research interest is "Modeling and designing resource efficient Distributed MANET Management System (DMMS) for multi-service applications". He got his Bachelor of Science degree in Computer Science (CS) from American International University Bangladesh (AIUB), Bangladesh in May 2006. After his B.Sc, he worked in software industries for 2 years. He also completed his Master's degree in Electrical Engineering (EE) from Tuskegee University (TU), AL in December 2010. In his Master's thesis, he worked with 'Coded Cooperative Diversity with Turbo Codes' and his advisor was Dr. Fan Jiang.

**Dhadesugoor R. Vaman** is Texas Instrument Endowed Chair Professor and Founding Director of ARO Center for Battlefield Communications (CeBCom) Research, ECE Department, Prairie View A&M University (PVAMU). He has more than 38 years of research experience in telecommunications and networking area. Currently, he has been working on the control based mobile ad hoc and sensor networks with emphasis on achieving bandwidth efficiency using KV transform coding; integrated power control, scheduling and routing in cluster based network architecture; QoS assurance for multi-service applications; and efficient network management. Prior to joining PVAMU, Dr. Vaman was the CEO of Megaxess (now restructured as MXC) which developed a business ISP product to offer differentiated QoS assured multiservices with dynamic bandwidth management and successfully deployed in several ISPs. Prior to being a CEO, Dr. Vaman was a Professor of EECS and founding Director of Advanced Telecommunications Institute, Stevens Institute of Technology (1984-1998); Member, Technology Staff in COMSAT (Currently Lockheed Martin) Laboratories (1981-84) and Network Analysis Corporation (CONTEL)(1979-81); Research Associate in Communications Laboratory, The City College of New York (1974-79); and Systems Engineer in Space Applications Center (Indian Space Research Organization) (1971-1974). He was also the Chairman of IEEE 802.9 ISLAN Standards Committee and made numerous technical contributions and produced 4 standards. Dr. Vaman has published over 200 papers in journals and conferences; widely lectured nationally and internationally; has been a key note speaker in many IEEE and other conferences, and industry forums. He has received numerous awards and patents, and many of his innovations have been successfully transferred to industry for developing commercial products.