Management of Quality of Service and other functions in mobile Ad Hoc networks

by
Grunde Eikenes and Ole Erik Grostøl

Masters Thesis in
Information and Communication Technology

Agder University College

Grimstad, May 2003

Abstract

This Masters thesis deals with management of mobile Ad Hoc networks, with main focus on QoS management. The needs for management of such networks and the challenges in managing them are an important part of the theoretical discussion. Different methods like SNMP, ANMP, Guerrilla management and Policy-based management are described and evaluated in order to be used in management of mobile Ad Hoc networks.

The task of managing a MANET is important in order to maintain an effective and stable network. Securing the transmission of data and avoiding unauthorized network users are important security tasks. Access to an updated topology map, viewing all the network nodes and their belonging links, are vital input for the manager in order to get total overview of the network. Configuring the network with adequate QoS will give increased quality for important traffic.

Considering the state of affairs on the different management methods and the time we had at our disposal, SNMP was the chosen management method for further testing. We focused on real network implementation and testing. The goal of the test was to prove the theoretical assumptions made on SNMP and to state its suitability for management of mobile Ad Hoc networks.

Based on theory and testing we found that SNMP is suited for small mobile Ad Hoc networks. Because of its centralized architecture and frequent polling of data, many network nodes will result in much overhead. Larger networks would benefit from a more event-based collection of information and clustering of the management nodes. Policy-based management together with ANMP seems like a good solution in managing mobile Ad Hoc networks, especially for QoS management.

Preface

This Masters thesis has been written in cooperation with the company Applica at Vigeland, Norway. The thesis represents a half years work and is a part of the Masters degree (Siv.ing) in Information and Communication Technology (ICT) at Agder University College.

From Applica we have had two supervisors, Erling Sandvik (Supervisor) and Berner Vegge (Co-supervisor). In addition, Geir Køien has been our supervisor from Agder University College. We would like to give great thanks to all the supervisors, and specially the help and support given by Erling and Berner at Applica. We would also like to express our thanks to Applica for giving us this opportunity.

Grunde Eikenes

Ole Erik Grostøl

Grimstad, Spring 2003

Table of contents

List of figures

List of tables

1 Introduction

1.1 Thesis introduction

Mobile Ad Hoc networks (MANETs) are multihop wireless networks. The nodes participating in a MANET operate both as end hosts and as routers, and they may be highly mobile.

The interest for MANETs has increased in the last few years. The intended range of use for such wireless networks varies from military operations to use in office environments. Common for MANETs are that they shall operate in places and situations where no network infrastructure is present. Reasons for lack of infrastructure can be difficult areas, lack of time, high costs or that the infrastructure has been destroyed. MANET [1] is a working group in the Internet Engineering Task Force (IETF) organisation, working on standardization of protocols in MANETs.

Management of MANETs has not been a prioritised problem area so far. However, there has been proposed a few interesting solutions. Ad Hoc Network Management Protocol (ANMP) [2] is a management protocol intended used in a MANET. This protocol is actually a development of the well known wired-based management protocol Simple Network Management Protocol (SNMP) [3]. The Guerrilla Management Architecture [4] proposes a management architecture that facilitates adaptive and autonomous management of MANETs. None of the mentioned management methods focus on Quality of Service (QoS). However, this is done in a study where Policy-based Management [5, 6, 7, 8] is intended used in MANETs. Work including Policy-based management has earlier mainly been focusing on fixed high-bandwidth networks.

It is important to evaluate the need for management of MANETs before considering how to perform the task of management. After evaluating the needs for management with main focus on QoS, we will look at the challenges in managing such a network. The characteristics of a MANET are quite different from a fixed wired network, resulting in new challenges that need to be overcome.

Based on the theoretical work on management in MANETs, we will evaluate the currently available management methods and propose one of them for further testing. The selected management method will be tested in real environments, and it will be evaluated in order to see how well it is suited for a MANET.

1.2 Task description

Thesis title
Management of QoS and other functions in MANETs

Background
A MANET consists of mobile nodes that can move freely in an arbitrarily manner, while still maintaining or establishing connections to other nodes. Such a network is independent of existing infrastructure. If two nodes are out of each others radio range, they can use other nodes as relay points for the connection. IETF is involved in the standardization work for such networks (routing protocols etc).
Some of the features that characterize such networks are dynamic topologies (host and/or network mobility), bandwidth-constrained variable capacity links, limited physical security and survivability, and nodes with limited battery life, processing power and storage capacity. These characteristics pose significant challenges to the management of MANETs.

Thesis subject definition
This Masters thesis shall evaluate needs and possible solutions for management in MANETs. The main focus shall be on QoS management, but other management aspects like configuration management, security management etc, shall also be considered. Both the management needed before a MANET is put into operation and the management needed during operation shall be considered.
A management system for relevant parameters shall, if possible, be implemented and tested in an Ad Hoc environment.

Suggested activities
1. Describe the characteristics of MANETs in general, including QoS in MANETs.
2. Discuss the needs for management in MANETs, with main focus on QoS management.
3. Discuss the different methods for management of a MANET (SNMP, ANMP etc).
4. Choose the most promising technique and implement a part of QoS management for a MANET, and test the implemented solution in an Ad Hoc test environment.
5. Discuss the experiences from theoretical input and tests and suggest one or more techniques for Ad Hoc management.

1.3 Report overview

The target group for this report comprises network engineers and students working with MANETs and in particular management of such networks. The report can also be of interest for anyone interested in the following topics: QoS, network management and MANETs.

In each of the three next chapters we give a general introduction to a central topic in this Masters thesis. This is necessary in order to learn or brush up the reader's knowledge of these topics.

Chapter 2 gives an introduction to MANETs. Typical characteristics of such networks and routing in MANETs are central subjects in this chapter. (Suggested activity 1)

QoS is the topic in chapter 3, where we describe the different approaches of QoS and then integrate this chapter with the previous. (Suggested activity 1)

Chapter 4 is the last of the introduction chapters, and deals with network management. The different management functional areas are an important part of this chapter. (Suggested activity 1)

In chapter 5 we make use of the theory in the three previous chapters, and discuss management in MANETs with main focus on QoS. Needs for management in a MANET both before it is put to work and during its lifetime is an essential part of the discussion, but also the different challenges in managing such a network. (Suggested activity 2)

Chapter 6 describes different management methods. Each of them is discussed in relation to the previous chapter in order to state how suited they are for use in MANETs. Based on the discussion we will choose one of the methods to be used for testing. (Suggested activity 3)

Chapter 7 gives a description of the test, which is divided into three parts. First we test a topology map, to see how fast the network topology can be updated. Next, we test the SNMP-protocol in order to measure its overhead and to see how it reacts to different traffic loads in the network. Finally we do a bottleneck test, to test the SNMP protocol under such circumstances. (Suggested activity 4)

In chapter 8 we make an overall discussion where we combine the discussions made in the three previous chapters, and finally in chapter 9 we make conclusions based on the discussions made throughout this report. (Suggested activity 5)

2 Mobile Ad Hoc networks

2.1 Introduction

In the resent years the development of wireless communication devices like Personal Digital Assistants (PDA), multifunctional phones, Personal Computers (PC) etc. have increased dramatically. We envision a widespread use of wireless transmission in the future, and the use of Internet Protocol (IP) would be a natural part of this. The overall goal of mobile Ad Hoc Networks (MANET) is to support resilient and effective wireless network by adding routing functionality to the mobile nodes. MANETs have a dynamic and rapidly changing multihop topology, and the wireless links connecting the network nodes have a limited bandwidth. The Network Working Group defines in RFC2501 [9] the MANET envision as :"The goal of mobile ad hoc networking is to extend mobility into the realm of autonomous, mobile, wireless domains, where a set of nodes – which may be combined routers and hosts—themselves form the network routing infrastructure in an ad hoc fashion"(Figure 1).

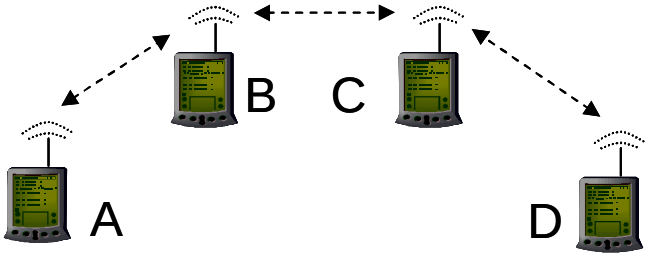

Figure 1 MANET network

2.1.1 Range of application

There is today a great need for MANET solutions, and it is expected to increase in the future. There is a commercial, industrial and military need for robust MANETs using mobile IP to run IP compatible applications. Solving problems like routing and clustering support etc. must however first be overcome before the use of MANETs can be widespread.

In light of the resent year nature disasters and terror attacks like the one to World Trade Centre 11 September 2001, where the whole communication infrastructure on Manhattan broke down, the work on MANET solutions for rescue scenarios really blossomed. In such scenarios there are a need for communication between different groups of people like fire departments, police departments and medical departments, and the ability for these groups to communicate inside the same networks.

Mobile solutions for military use have been under development since the middle of the seventies. In light of new radio technology there has been a movement against using wireless multihop networks like MANETs. Security tasks are of high priority in such military scenarios. The ability to quickly set up communication channels in the battlefield and exchange data etc. could make the difference for the outcome of the battle.

Other scenarios where MANETs could be of great utility value are sports arrangements and office scenarios.

2.2 Characteristics of MANETs

A MANET consists of wireless communication devices which are free to move in an arbitrarily way. The nodes may be located in or on people, trucks, cars, or very small devices. A MANET is an autonomous system of mobile nodes. The network may operate in isolation, or it may have gateways to a fixed network

There are a few distinctive characteristics of MANETs [9]:

1) Dynamic topology: The nodes mobility introduces dynamic topology. The movement of the network nodes are random and may be highly rapid. This makes the task of routing traffic through the network, in order to establish point-to-point connections very challenging.

2) Bandwidth: Wired networks have a significant higher bandwidth than wireless network solutions. In addition, the experienced available bandwidth of the links during multiple access, fading, noise and interference conditions is way below the theoretical available bandwidth. One effect of the low link capacity is that congestion is the norm rather than the exception. This may in worst cases lead to a network collapse.

3) Energy: The power supply of the nodes participating in MANETs will be batteries or other fast discharging power supplies. The development of MANETs must therefore take this limited resource into consideration with a view to limit the traffic in each of the nodes to save battery. Nodes running on batteries will also result in a more dynamic topology.

4) Limited physical security: MANETs are more vulnerable to physical threats than regular wired networks. The increased possibility for eavesdropping and spoofing should be considered.

2.3 Routing in MANETs

As described in [10, 11] there is no pre-existing infrastructure in MANETs. When nodes are not in direct contact with each other, there would be a need for nearby nodes to relay packets. Routing packets among any pair of mobile nodes become a great challenge because they can move around randomly. A path between two nodes may be of good quality one moment, and then for the next moment not work at all. In MANETs every node will be a potential router. The most salient problems for routing in MANETs are limited bandwidth and rapid topology changes which may cause link failures. The intention for MANETs is to use routing protocols based on the Internet Protocol (IP).

There are several desirable qualitative properties in MANETs [9]:

Loop freedom, which avoids the problem of packets flowing in the network for a long time of period or for ever.

Reactive operation, which sets up routes exclusively when they are needed, and maintained as long as the communication goes on. This solution avoids the constant flooding of routing information, and therefore at all time has updated routes at dispossession. It rather uses routes that all ready have been set up or is in the process of being set up. The reactive protocols utilize the bandwidth economically, but have a tendency to suffer from higher route discovery delay.

Proactive operation, which obtain routes to all nodes in the network even if there is no traffic transmission going on. The routing tables change because of topology changes and not because of the traffic transmitted. Each node is periodically sending out control messages so that each node has the complete network routes. The dynamics of the network has to be considered in order to calibrate the transmission of control messages so that the routes are valid. The proactive way of routing would waste a lot of bandwidth. This is due to the fact that all routes are maintained whether they are used or not. However, transmission of traffic would not suffer from high delay because of setting up new routes before every transmission.

Security, with lack of this property function on network- or link layer, MANETs would be very vulnerable to manipulation of packets, snooping traffic etc. Such security functions already exist in wired networks, but they are harder to obtain in MANETs due to the “physical” security.

Sleep period, most mobile nodes goes into a sleep mode when they are not used for a period of time to conserve energy. In such occasions, MANETs will loose contact with the sleeping nodes until they are active again. MANETs need routing protocols that handle this problem without too unfortunate consequences.

2.4 Physical layer

The spreading of a network technology is connected to an organisations ability to develop good network solutions to a low and competitive price. Today there are two widespread Ad Hoc wireless technologies for commercial sale, 802.11b and Bluetooth. 802.11b is suited for large Wireless Local Area Networks (WLAN), and the transmission range between two nodes are about 100 metres. Bluetooth is made for smaller networks, and its corresponding transmission range is about 10 metres.

2.4.1 802.11b

802.11b [12] operates in the 2.4 GHz spectrum, and the physical layer is an extension of the 802.11 physical layer, which supports 1 and 2 Mb/s. 802.11b can support higher data rates equal to 5.5 and 11 Mb/s with the use of Complementary Code Keying (CCK) with Quadrature Phase Shift Keying (QPSK) modulation and Direct Sequence Spread Spectrum (DSSS) technology. By using dynamic rate shifting, data rates are allowed to automatically adjust due to how noisy the transmission conditions are. This means that bad conditions results in low bandwidth, while good conditions automatically results in a higher bandwidth.

The MAC sublayer operates as the interface that connects the physical layer to the host devices, and it supports two different modes, Ad Hoc and infrastructure. Two important functions on the MAC layer are Cyclic Redundancy Check (CRC) and Packet Fragmentation. The use of fragmentation reduces the need for retransmissions, because the probability for errors increases with a larger packet size. Also, when packets get corrupted it will only be necessary to retransmit the corrupted fragment. 802.11b uses Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA). This includes that each network node listens to the traffic sent from the other nodes. A node can only send packets when an idle transmission channel is detected.

Figure 2 shows the different modes of 802.11b. The infrastructure mode is most used in office environment etc, while the Ad Hoc mode is relevant for MANETs.

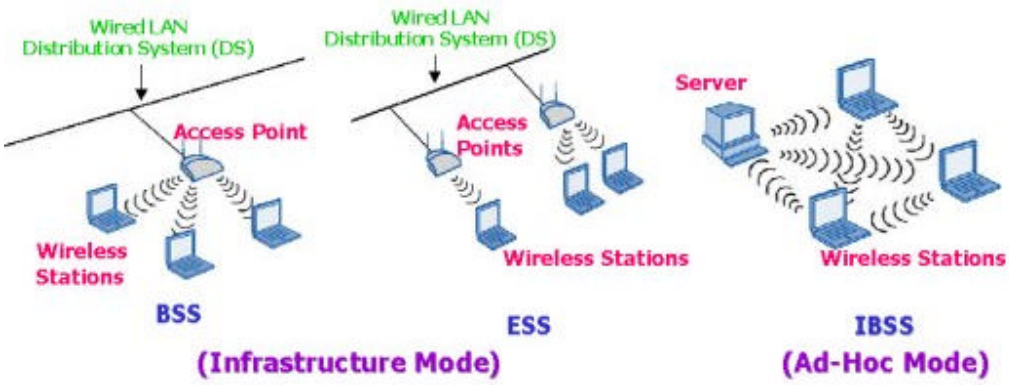

Figure 2 802.11b modes

2.4.1.1 Bluetooth

The main goal for the Bluetooth technology [13] is to remove the cable between different devices, like communication between printer and computer or between phone and headset etc. The technology may also function in a point to multipoint fashion, Ad Hoc. Bluetooth has a very limited LAN ability in form of a Personal Area Network (PAN).

The Bluetooth technology also operates in the 2.4 GHz band, and uses Gaussian Frequency Shift Keying (GFSK) modulation scheme and Frequency Hopping Spread Spectrum (FHSS). It is theoretical possible to achieve a range up to 100 metres, but most devices will deliver a max transmission range up to 10 metres. Under optimal condition, Bluetooth will be able to deliver up to 1 Mbps

3 Quality of Service

3.1 What is QoS?

The United Nations Consultative Committee for International Telephony and Telegraphy (CCITT) Recommendation E.800 has defined a widely accepted definition of QoS as: "The collective effect of service performance which determines the degree of satisfaction of a user of the service". QoS is a measure of the quality a user can expect from a given service provided by a network. There are a variety of applications available in the market today, which all have their own specific demands to the QoS. There must be an interaction between what the applications demand from the network and the services that the network provides.

3.2 QoS parameters

The different QoS demands from applications can be summarized in the following parameters:

Bandwidth/Throughput – Applications have different needs for bandwidth and requires a certain amount of rate to be offered by the network. The bandwidth available in the network is finite, so the different applications compete for the available bandwidth resources. Throughput is the actual rate offered by the network, and hopefully this satisfies the bandwidth required for the specific application.

Latency – From an application point of view, latency is the delay that it can tolerate in delivering a packet of data. Another point of view is the amount of time it takes for a packet of data sent from a source port until it reaches its destination port. Latency accounts for transit time as well as queuing and processing delays in reading header information and acting on it.

Jitter – In short jitter is the variation in latency. Common jitter-sensitive applications are streaming video and voice.

Packet loss – While traversing a network, packets of data may pass through multiple processing stages as they are routed from source to destination. Bottlenecks may arise, resulting that packets are filling queues faster than they are being forwarded out. For the network to remain stable, the only solution is to drop packets (packet loss).

The QoS parameters defined above are a measurement on how efficient and reliable a network is.

3.3 QoS forwarding mechanisms

The forwarding process is taking place in a router when it receives a packet on one of its input interfaces. The router then reads the packet header and gives it a proper treatment, before it is sent out on the correct output interface. The treatment of a packet varies from router to router, due to what is installed in its forwarding engine. A router supporting QoS will typically contain the following mechanisms in its forwarding engine:

Admission control – this mechanism is used to determine if a packet flow (belonging to a particular node and application) is able to get its requested resources. The packet flow is refused if there are not enough available resources in the router.

Packet classification – is to determine a packet's needs and rights for treatment in a router. Different factors may have influence on the classification of a packet. Transport protocol (TCP, UDP), incoming interface, packet size, source- and destination address are some examples. Packet classification will have influence on both policing & marking and queuing & scheduling.

Policing & marking – Policing is a process of ensuring that incoming traffic belonging to a given class is conforming to the traffic profile defined for that class. If a packet is out of profile, one policy might be to drop the packet. Another policy might be to mark the packet with lower priority in the scheduler. Marking packets is a way to treat them differently. For instance, different marking is a way to tell what packets to drop first if the network is heavily congested.

Queuing & scheduling – Queuing is the process of passing the packets out to the output queues. It also determines how packets are dropped when congestion occurs. Scheduling implies how the packets in the output queues are sent out to the different interfaces. A scheduler shall try to provide the different packets their promised resources. There are many ways to manage output queues. The simplest one is a First In, First Out (FIFO) queue, where the first packet to arrive is the first packet to leave. This approach is suited for links with minimal congestion. Other queuing algorithms are Fair Queuing (FQ) - one separate queue for each flow, and the queues are served in a round robin manner, Weighted Fair Queuing (WFQ) - one queue for each flow and give priority to low bandwidth flows, Priority Queuing (PQ) - serves always the queue with highest priority, and Class Based Queuing (CBQ) - reserves a portion of the link bandwidth for each selected traffic type. Random Early Detection (RED) and RED with In and Out (RIO) are mechanisms to avoid congestion, e.g. queues are being filled up. The first one drops packets randomly as the queues begins to fill up, in purpose to tell Transport Control Protocol (TCP) connections to slow down. The other one does in addition some sort of drop precedence.

3.4 QoS approaches

3.4.1 Best effort services

The best effort services are the simplest of the three QoS approaches. "Best effort" means that the network tries to deliver a packet in a best possible manner, but if it fails to deliver a packet or if a packet gets lost or corrupted, the network does nothing to fix the problem.

The best effort services offer basically no quality control. FIFO queues have been used to handle traffic, but implementation of queuing algorithms, such as CBQ and WFQ can give an increased QoS.

The growth in real time applications puts new requirement to the QoS to be provided by the network. For instance, Voice over IP (VoIP) is not working well with high delay and jitter. The best effort services can not guarantee such QoS parameters, and other services are therefore developed to meet the requirements for such applications.

3.4.2 Differentiated services

The differentiated services (DiffServ) [14] divide traffic into a small number of traffic classes. All the traffic is sorted among these different traffic classes, and the routers treat each traffic class differently. As an example: VoIP traffic belongs to traffic class 1, and email traffic belongs to traffic class 2. Traffic class 1 is given higher priority by the routers in order to avoid much delay, which is a critical factor for such traffic.

The architecture of DiffServ distinguishes between edge routers and core routers. The edge routers are located at network boundaries, while the core routers only communicate with nodes within the network. The goal is to keep the core routers simple and place the complex functions to the edge routers. By doing this, the architecture will be more scalable. Also, DiffServ requires no signalling. Edge routers may also be divided into ingress- and egress nodes if traffic is flowing between two different DiffServ domains. Traffic entering a DiffServ domain passes through an ingress node, while traffic exiting a DiffServ domain passes through an egress node.

3.4.3 Integrated services

Integrated services (IntServ) [15] are strongly connected to the Resource Reservation Protocol (RSVP). IntServ handles classification and marking of packets, while RSVP is used for signalling to reserve desired resources in the routers along the path between the sending node and the receiving node.

IntServ offers two service classes in addition to best effort, respectively Controlled Load and Guaranteed Service. The difference between them is that Guaranteed Service shall provide exact guarantees to a packet flow, while the Controlled Load only gives an assurance that a very high percentage of the packets will have their requirements to QoS fulfilled.

3.5 QoS on different layers of the Open Systems Interconnection (OSI) model

QoS support in MANETs affect most of the layers in the OSI-model. Physical layer, link layer, network layer, transport layer and application layer have all influence when supporting QoS across the protocol layers.

3.5.1 Physical layer

Since the transmission of data in MANETs are wireless, the transmission channels are time-varying. This is due to the problems introduced by the channel fading, multipath fading and mobility. To support QoS in such environments, channel estimation will therefore be an important task in order to synchronise the receiver and the transmitter. AVLSI [16] is a research group working among others with adaptive techniques in MANETs, including adaptive channel state information.

Compared to wired networks, noise and collision will be a much bigger problem in wireless networks, making the QoS provisioning even harder.

3.5.2 Data link layer

Because of synchronization problems in wireless networks, synchronous MAC-protocols like Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) are not suitable. Asynchronous MAC-protocols therefore needs to be used with a more distributed control mechanism. Most such MAC protocols are designed for a single-hop wireless network, but these protocols does not allow for the hidden terminal problem. This problem occurs when two nodes, which are out of transmission range of each other, are sending to the same node and causing a collision. However, the IEEE 802.11 Distributed Control Function (DCF) solves the hidden terminal problem entirely, using its Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [17].

Another problem remaining is to provide real-time traffic support. This is not done by the 802.11 DCF, which only supports best effort service. Recently, many MAC schemes have been proposed in order to provide QoS guarantee for real-time traffic. An extension to the 802.11 DCF and a scheme called Black Burst have both proposed service differentiation, giving real-time traffic higher priority over other traffic, which are treated as best effort. There has also been proposed a scheme providing guaranteed bandwidth

support for real-time traffic, called Multihop Access Collision Avoidance with Piggyback Reservation (MACA/PR). IEEE802.11 has a family of different WLAN standards, where 802.11e defines an enhancement of MAC to support LAN applications with QoS. The new QoS mechanisms in 802.11e are Enhanced Distributed Coordination Function (EDCF) and Hybrid Coordination Function (HCF)

3.5.3 Network layer

Most routing protocols proposed for MANETs today do not take QoS into consideration. In most cases messages are routed the shortest available path, which may not be adequate for applications that require QoS support. For instance, an application requires a certain amount of bandwidth. The network is only able to provide this amount of bandwidth through certain nodes, which may not be the shortest available path. A routing protocol that is based on the desired QoS can be termed as QoS aware. “The primary goal of the QoS-aware routing protocol is to determine a path from a source to the destination that satisfies the needs of the desired QoS” [17].

There has been proposed a few QoS-aware routing protocols, that find a path based on the desired QoS (particular based on bandwidth or delay). Core Extraction Distributed Ad hoc Routing (CEDAR) algorithm is an example of a routing scheme based on available bandwidth.

Routing is just one aspect of providing QoS on the network layers. Another aspect is to offer services that work better than the “best effort” approach. This approach must take into consideration the different characteristics of MANETs like dynamic topology and variable link quality.

The IntServ/RSVP approach is not very suitable to MANETs because of their limited resources and dynamic topology. IntServ/RSVP requires huge storage and processing overhead for each mobile host, and this increases proportionally with the number of flows. The mobile nature of MANETs also makes reservations of guaranteed services very difficult.

DiffServ does not load the network with much overhead, because individual flows are aggregated into set of flows, and no reservation signalling is necessary. This makes the task of routing simpler, and could therefore be a good solution for MANETs. However, the problem with DiffServ in MANETs is that there is no clear definition of which nodes that are core- and edge routers, due to the dynamic topology.

Flexible QoS Model for MANETs (FQMM) [18] is a QoS Model proposed for MANETs. The basic idea of this model is to combine the per-flow property of IntServ and the service differentiation of DiffServ. This means that the highest priority classes are given per-flow treatment, while the other priority classes are given per-class treatment. The FQMM approach defines three types of nodes (ingress, core and egress), just like in DiffServ. The difference is that the different nodes in FQMM have nothing to do with

their physical location. A node is characterized as ingress if it is transmitting data, core if it is forwarding data and egress if it is receiving data.

3.5.4 Transport layer

Traffic is transported by either the User Datagram Protocol (UDP) or the TCP-protocol. UDP is used by some real-time applications and requires little functionality from the network. TCP on the other hand offers reliable end-to-end packet delivery and guaranteed in-order packet delivery. One problem with the TCP-protocol to be used in MANETs is the one that TCP assumes that most packet losses are due to network congestion. Of course this happens in MANETs, but most packet losses are likely to occur because of channel noise and route changes. TCP will activate its congestion control and avoidance algorithm whenever the TCP-sender detects a packet loss, and the end-to-end throughput will therefore be very poor. There are work going on to improve the TCP performance in MANETs. This implies that error losses need to be distinguished from congestion losses.

3.5.5 Application layer

It is very difficult to provide QoS in a highly dynamic environment. The applications therefore need to adapt to this reality. One solution is to specify a range of values that they can tolerate. The network tries to provide resources within this range, and the applications must be able to work under such circumstances.

3.5.6 Inter-layer design approaches

In addition to the work that has been done on each individual layer, there have also been proposed inter-layer QoS frameworks for MANETs. Three of these attempts are INSIGNIA [19], Stateless Wireless Ad Hoc Networks (SWAN) [20] and integrated Mobile Ad-hoc QoS framework (iMAQ) [21].

The primary goal of INSIGNIA is to provide adaptive QoS to real time traffic, with minimum signalling. The framework is not bounded to a specific routing protocol and uses in-band signalling. This type of signalling is carried along with the data packets. The contrast to in-band signalling is out-of-band signalling, where the signalling uses explicit control packets. RSVP is an example of the last one. In-band signalling is the most suited for MANETs, due to their bandwidth and power constraints and dynamic topology [22]. INSIGNIA also uses soft-state resource management, which is more flexible to use in MANETs compared to hard-state [22]. A soft-state approach include that the intermediate routers reserves resources for a time of period. If no more packets associated with this reservation arrive within this time of period, the resources are released.

The INSIGNIA QoS framework (figure 3) allows applications to specify desired bandwidth range and helps in resource allocation, restoration control and session

adaptation between the communicating mobile hosts. The in-band signalling establishes, restores, adapts and tears down adaptive resources between source-destination pairs. Admission control is responsible for allocation of bandwidth to the different flows. Packet forwarding classifies the incoming packets forwards them to the appropriate module, e.g. the signalling module. Routing keeps an updated topology so that the packet forwarding always has an updated topology available. Packet scheduling responds to location-dependent channel conditions when scheduling packets in wireless networks. The INSIGNIA QoS framework is designed to work over multiple link layer technologies, but the provisioning of QoS is strongly connected to the QoS-support of this layer.

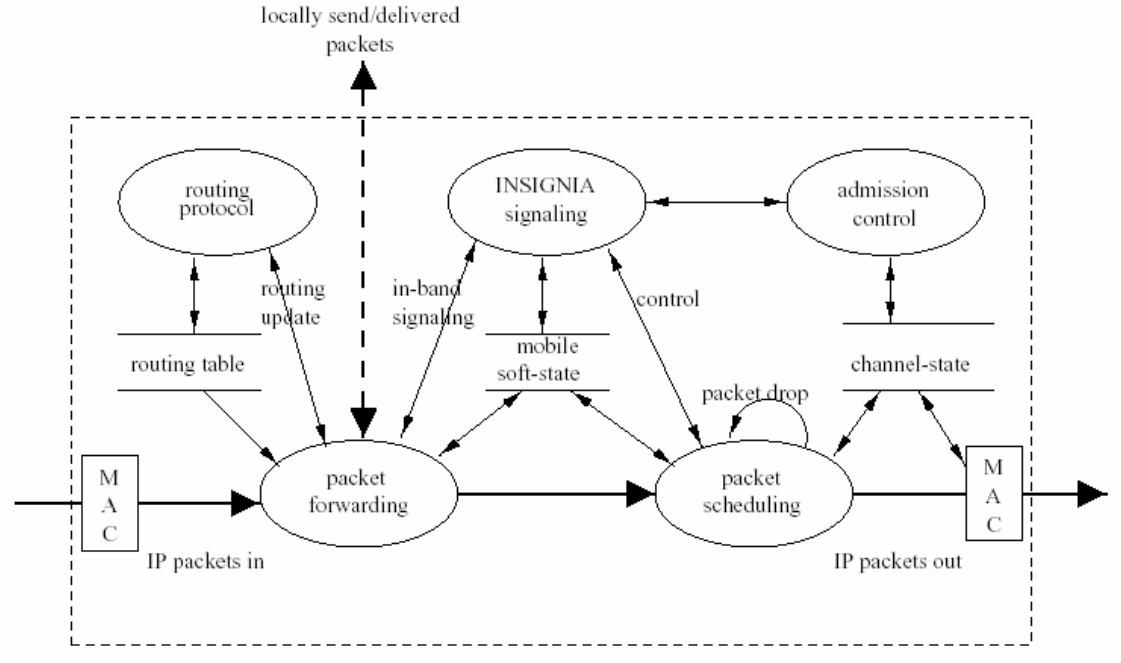

Figure 3 INSIGNIA QoS framework [19]

SWAN is "a stateless network model which uses distributed control algorithms to deliver service differentiation in mobile wireless ad hoc networks in a simple, scalable and robust manner." [20] Stateless means that there is no need for signalling and state control mechanisms to establish, update, refresh and remove per-flow state, which is done in "stateful" QoS approaches like INSIGNIA. SWAN uses local rate control for UDP- and TCP best-effort traffic, and sender-based admission control for UDP real-time traffic. Explicit congestion notification (ECN) is used to dynamically regulate admitted real-time sessions when the topology changes. SWAN can operate over best-effort MAC-technology, and does not require QoS-capable MAC-technology to deliver service differentiation.

The SWAN model (figure 4) includes many mechanisms to support rate regulation of best-effort traffic. Between the IP-layer and the best-effort MAC-layer, there operates a classifier and a shaper. The classifier is capable of differentiating traffic, while the shaper

shall delay best-effort packets in conformance with the rate calculated by the rate controller. The admission of new real-time sessions is only decided in the source nodes admission controllers. This is based on the result of an end-to-end request/response probe, typically sent at the beginning of a session. The probe shall estimate the local bandwidth availability.

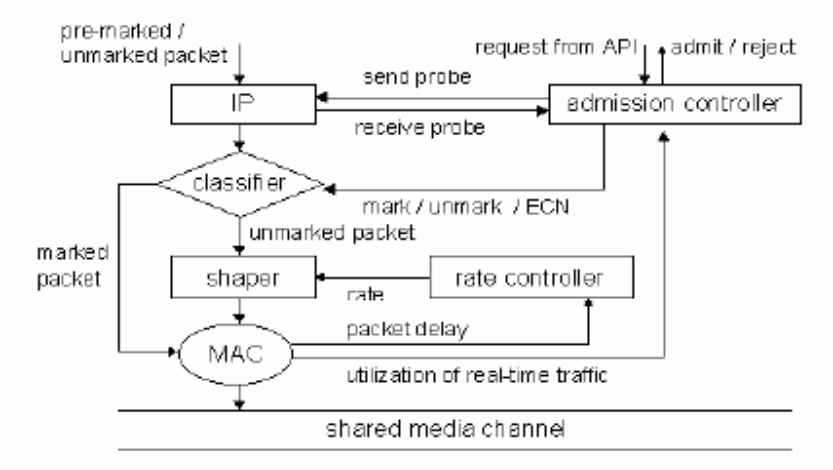

Figure 4 The SWAN model [20]

iMAQ supports transmission of multimedia data over a MANET. The framework (figure 5) includes an Ad Hoc routing layer and a middleware service layer. In each node these two layers share information and communicate in order to provide QoS assurance to multimedia traffic. The routing protocol is predictive location-based and QoS-aware. The middleware layer also communicates with the application layer.

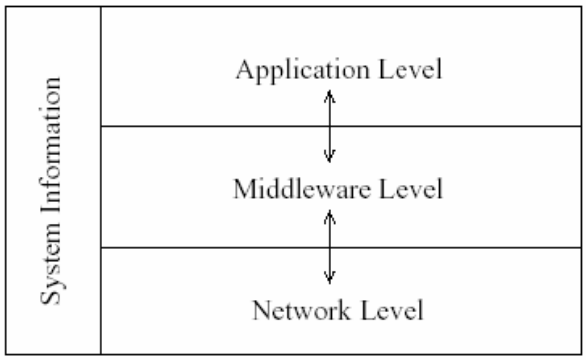

Figure 5 The iMAQ framework model [17]

3.6 QoS in MANETs

Because of the characteristics for MANETs, QoS provisioning is an even harder task then in normal wired networks [17].

- The bandwidth is limited, making it very difficult to satisfy all network-users, especially those running highly bandwidth demanding applications.
- Wireless transmission of data results in a much higher percentage of errors in the transmitted packets, due to fading, multipath etc. This makes the QoS provisioning to error-sensitive application more difficult and measures of QoS parameters very unpredictable.
- The hidden terminal problem is also introduced with MANETs, which results in more collisions and degraded network utilization.
- Because of the mobility of the network nodes, the topology will often change and new links will constantly be activated and deactivated. Together with the variable link quality, this makes a precise maintenance of network state information very difficult and therefore a challenging task for the routing algorithms. Established routing paths may be broken during the process of transferring and new routes needs to be established quickly. Reservation of resources in routers is therefore a problem, because the new routers may not be able to provide the same quality.
- Mobile devices have limited power supply, due to nodes running on batteries. This result in more frequent topology changes and must be taken into consideration when resources is allocated. Routing much traffic through a node with low battery power may not be a good idea. Also, all the techniques for QoS provisioning should be power-aware and power-efficient.
- MANETs are among others intended used in hostile environment. Security is therefore important in order to deny intruders in sabotaging the network, and hence degrade the network performance. Wireless traffic is also more insecure which increases the difficulties in providing a secure network.

4 Network management

4.1 What is network management?

"In general, network management is a service that employs a variety of tools, applications, and devices to assist human network managers in monitoring and maintaining networks." [23] A network is a complex system, consisting of a great number of nodes and a variety of protocols running on them. Each node keeps track of a lot of information that needs to be maintained and manipulated, for instance, routing tables, traffic flows, etc. Network management will typically involve monitoring and controlling this information stored in every node in the network.

4.2 Management functional areas

The International Standards Organization (ISO) has developed a network management model, that most network management systems today address. The model consists of five functional management areas, where FCAPS is the acronym of these functional areas. (**FCAPS** = **F**ault, **C**onfiguration, **A**ccounting, **P**erformance and **S**ecurity). However, most network management implementations don't really cover all of these areas.

4.2.1 Fault management

"The goal of fault management is to detect, log, notify users of, and (to the extent possible) automatically fix network problems to keep the network running effectively." [23] Faults are divided into two categories, either a fault is a transient event or it is a persistent event. Transient events do not require any management corrections, but it is desirable to log this type of events. On the other hand, persistent faults do need to be corrected. Some persistent faults are corrected automatically, while others have to be corrected on the administration level. Automatically correction of persistent faults could be done either with initiative from the network manager (polling) or with initiative from the managed devices (indirect polling). The last one is most efficient in large networks, because it reduces the network traffic and processing load on the manager. Faults are often related to a root fault, it is therefore desirable only to report of the root fault, because of reducing the work to be done by the manager.

4.2.2 Configuration management

"Configuration management is the process of obtaining data from the network and using that data to manage the setup of all network devices." [24] Each network device keeps track of a lot of configuration information like different versions on hardware and

software, and this information is stored in databases in configuration management subsystems. Configuration management includes controlling and updating this type of information, and it can be a valuable source when a problem occurs.

4.2.3 Accounting management

"Accounting management involves tracking service usage and informing relevant users and authorities about the usage of resources and the costs associated with their usage." [25] This is relevant in billing the users of a network, but also for controlling the use of resources and the possibility to allocate these resources to different users.

4.2.4 Performance management

"The goal of performance management is to measure and make available various aspects of network performance so that internetwork performance can be maintained at an acceptable level." [23] Performance variables of interest will typically be, network throughput, user response time and line utilization. Managing these performance variables includes three main steps. First, the information on the different variables of interest is gathered. The next step is to monitor these variables to find a normal level. The final step includes setting different thresholds on the variables. Exceeding one of these thresholds can result in an alarm being sent from the discovering node to the network management system.

4.2.5 Security management

"The goal of security management is to control access to network resources according to local guidelines so that the network cannot be sabotaged (intentionally or unintentionally) and sensitive information cannot be accessed by those without appropriate authorization." [23] Typically, a security management subsystem will monitor users logging on to different network resources, and accept or deny according to if the access code was appropriate. Other important functions that the security management should provide are confidentiality, data integrity and audit ability.

4.3 Management architectures

4.3.1 Overview

First in this chapter we will introduce a basic management architecture, where the basic principles of doing network management are explained. The three next chapters describe the three most common network management architectures, respectively centralized, hierarchical and distributed, described in [24]. Finally we will describe a management architecture introduced by the work of Policy-based management.

4.3.2 Basic management architecture

The architecture of network management systems are generally the same. They share the same basic structure and set of relationships. End stations (managed devices) such as computer systems and other network devices, run software that enables them to send alerts when they recognize problems. For instance, an alert could be provoked by an exceeded user-determined threshold. When the management entity receives this alert, it reacts due to how it is programmed to react for this type of alert. Examples of actions to take place are; operator notification, event logging, system shutdown and automatic attempts to repair the system.

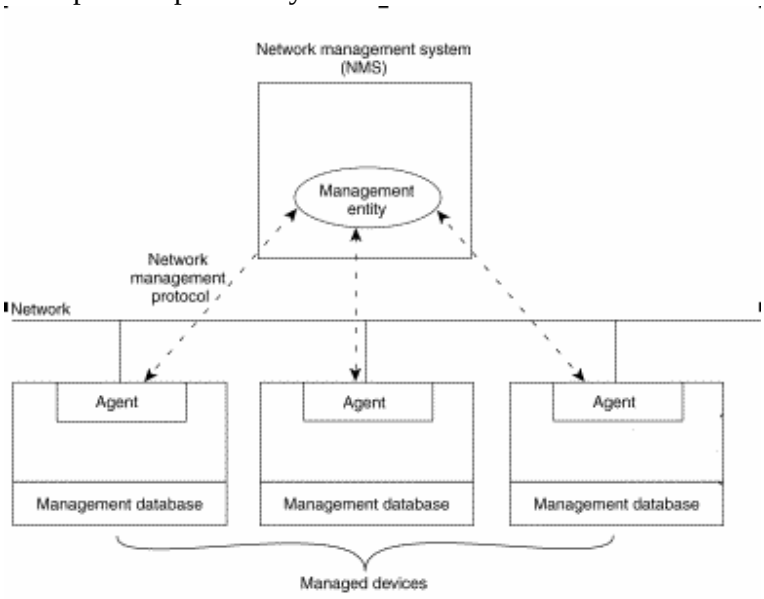

Figure 6 Typical network management architecture [23]

Management entities also have the possibility to poll end stations to set or get variables stored in their management database. There are two ways of how polling can be done, either automatic or user-initiated. Each end device has an agent which responds to both these types of polling. Agents are software modules that serve different tasks. First of all

an agent must compile information about the managed device in which it resides. This information must then be stored in a management database at the same managed device. As mentioned, this information could either be polled from a management entity (proactive) or sent to a management entity, initiated from an end station (reactive). The management entities and the managed devices are all part of a Network Management System (NMS), and information exchanged in a NMS is possible via a network management protocol.

4.3.3 Centralized architecture

A centralized architecture has the network management platform on one computer system. This computer system is responsible for all network management duties on all network devices and has one centralized database containing the network management information. The functions of the single management system are as follows:

- ? Handle all network alerts and events
- ? Keeping all the network information
- ? Accessing all the management applications

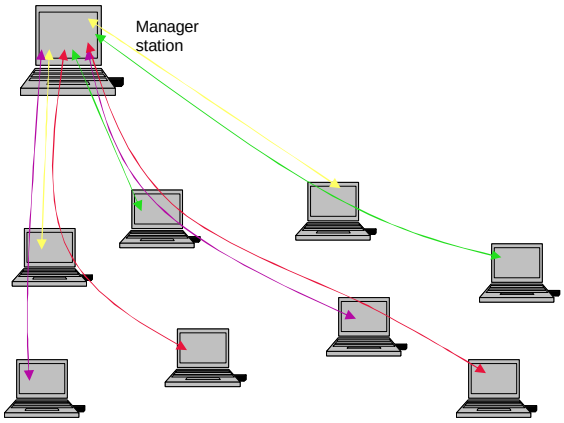

Figure 7 Centralized architecture

4.3.4 Hierarchical architecture

A hierarchical architecture uses multiple systems, with one system acting as a central server and the others working as clients. By organizing the architecture this way, some functions of the network management may run within the server, while other functions may run within the clients. The server will typically control the other clients, which again control different parts of the network. The clients will not have their own database system, but use client/server database technology to access a centralized server database. The key features of a hierarchical architecture are as follows:

- ? It is not dependent of a single system

- ? Distributed network management tasks
- ? Centralized information storage

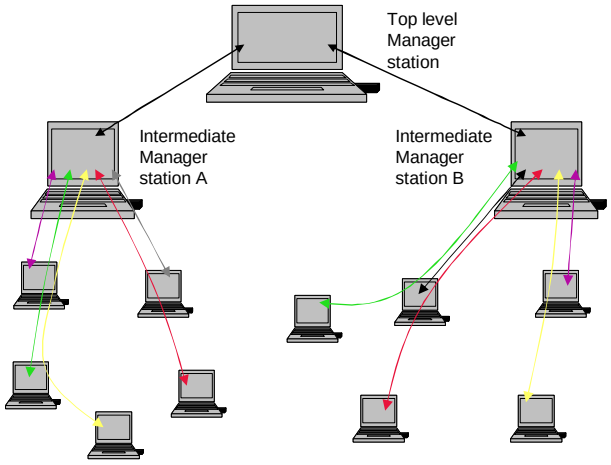

Figure 8 Hierarchical architecture

4.3.5 Distributed architecture

The distributed architecture combines the centralized and hierarchical approaches, and uses multiple peer NMS's. One of the NMS's is the leader, but each of the NMS's can have a complete database for all the network devices. The different databases need to be synchronized using database replication server technology. The distributed architecture will have the advantages of both the other approaches, including:

- ? Single location for all network information, alerts and events
- ? Single location to access all management applications
- ? Not dependent on a single system
- ? Distributed network management tasks

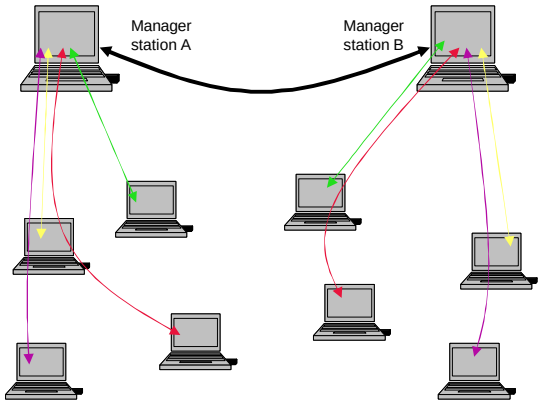

Figure 9 Distributed architecture

4.3.6 Policy-based architecture

Policies are plans for a network on how it can achieve its goals. Policies involve a set of rules to manipulate behaviour of the network and specifications on how actions should be executed. These policies are divided into two main groups. High level policies concentrate on general network goals, while low level policies concentrate on node level goals.

Policy Management Tool (PMT) gives the network manager a graphical interface so that he/she can interact with the network. The PMT is defining all the policies that shall apply for the network. PMT also validates the policies defined by the manager, and check if they are compatible to carry out in the network etc. The defined policies are then stored in a policy repository database. The Policy Decision Point (PDP) mainly gets policies from the policy repository and translates them into complex formats that shall be used to configure the Policy Enforcement Points (PEP). The PDP also has to monitor if new updates or changes of policies have been made on the PMT and stored in the policy repository. PMT does not have control over the policy status on the lower layer nodes. PEPs are the devices where the policies are implemented, and PEPs also need to announce to the PDP when it updates policies or when it receives a new one. This is done to obtain a desired autonomous system.

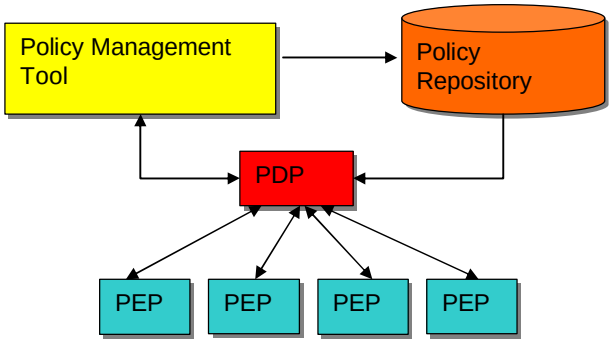

Figure 10 Policy-based architecture

5 Management in MANETs

5.1 Overview

In the previous chapters we have been given an introduction to the three subjects, MANET, QoS, and network management. In this chapter we want to melt these three subjects into each other and present a thorough analysis of challenges and needs for management in MANETs with main focus on QoS. This is according to activity 2 in the task description (chapter 1.2).

We first want to introduce an example scenario in 5.2, where a MANET is intended used after an earthquake. This is done to motivate for the needs of management in a MANET, which is discussed in 5.3 based on the FCAPS-model. In 5.4 we look at the different importance of management, dependent on where the MANET is intended used. There are a lot of challenges involved in the management task of a MANET. This is discussed in 5.5. We end the chapter by describing different management architectures and discussing their fitness for MANETs.

5.2 Introduction

To illustrate the needs for management in a MANET, we will introduce an example scenario. A big city has been exposed to an earthquake. Several buildings have collapsed, streets have been blockaded, the number injured and dead persons are large and the communication infrastructure has been damaged. Police forces, medical personnel, fire brigades and volunteers are coming to the accident area. The operation is organized in a hierarchic way, with head leaders in all rescue departments and other personnel with extended responsibility. The equipment they use changes in complexity, from small sensors to palmtops and fully equipped laptops, and the applications running on them have different requirement to the network they are operating in. A MANET (or more) has been established to handle all the communication, because of the missing infrastructure. It is important that the communication-possibility in the spontaneous network satisfies the demands from the variety of applications in a biggest possible way.

During the rescue operation, there are several different needs for communication. The large number of persons taking part in the operation put demands to the organization, and communication is the tool for this task. Because of the chaotic situation, there will continually be discovered new injured persons, damaged buildings, and areas to fence etc. Rescue personnel in these situations are crucial, and communication is needed to gather right and enough competent personnel. For medical personnel it might be desirable to download information about injured persons, in order to collect essential information about blood group, diseases etc. Contact between hospitals and medical personnel in the rescue operation are also important, so that the hospitals can prepare themselves for what they might expect, regarding to number of injured, type of injury etc. It is desirable that the different rescue departments are able to communicate within the same MANET. Communication must also be able to take place only within a rescue department, or other specified groups.

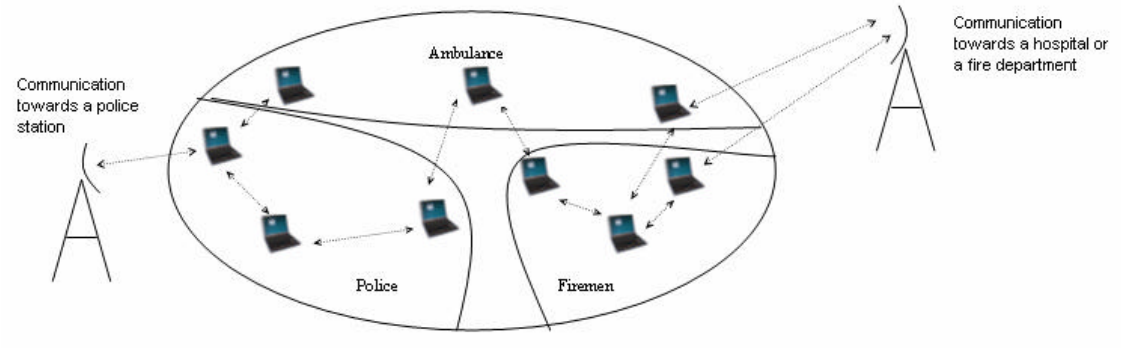

Figure 11 Different units connected to the same MANET

A scenario like this is a challenging task to get some sort of control and overview of, and this is where management of the network plays an essential role. A topology map with overview of all those nodes taking part in the rescue operation, together with a city map

would be of invaluable importance for the management part and for the rescue operation. This is because of the importance of a well functioning communication possibility through the whole operation, and within the whole accident area. The management node, with a topology map, will be able to discover parts of the accident area with no communication possibility, and choose necessary actions to solve the problem. Another problem that is likely to arise is the one of bottlenecks. A central node may be a relay point for too much traffic and the communication will dramatically degrade, due to high packet loss and delay. A discovery of this problem in a management node could be fixed by sending out another relay node to take over some of the traffic.

There will be different needs to the network dependent on the application to be used. A voice conversation will be useless if the delay and jitter through the network are above a certain threshold. Compared to a file transferring, where delay and jitter doesn't cause that big concerns, it is obvious that some QoS would be desirable. In an accident area as described above, the communication is wireless and nodes present are competing for the same limited bandwidth. A challenging management task will therefore concern providing the necessary requirement to different applications, including demands to bandwidth, delay, jitter and loss.

The communication sent between different nodes will be of highly confidential character, due to all the patient information. The press and other random persons should not be able to listen to the communication. The security in such a network will therefore be of high importance. Access control to the network and its resources will be a part of the security mechanism and a task for the management of a MANET. Undesirable participants to the MANET will also eat from the limited bandwidth and degrade the performance of the network.

The equipment used in a rescue operation described above, need some configuration and software/hardware installation to be done before it is operational. This type of pre-management is totally necessary, to get the nodes forming a network and route the traffic from a source to a particular destination. It could also be desirable to group different rescue departments, and put restrictions and possibilities to each group. For instance, a policeman belonging to one group is allowed to communicate with other policemen in the same group, but is not able to listen to communication between medical personnel in another group. It could be desirable to change the group settings during the rescue operation, which involves actions to be done on the management side.

5.3 Needs for management

The needs for network management in MANETs can largely be divided into two main tasks: monitoring the network and controlling the network. Monitoring the network typically involves tasks where information needs to be collected from the other nodes in the network. Keeping an up to date topology map, discover failure on network nodes, keeping track of network utilization, collecting information on users and applications are examples on monitoring tasks. Network control typically involves tasks where information is controlled or configured, for instance a security mechanism to control the access to a network resource or configuring a network node during an operation. Both monitoring and network control will be dependent on each other.

A more detailed division of the management tasks in a network is the one done by the ISO – the FCAPS model described in chapter 5.2. This model will be used while exploring the needs for management in a MANET later in this chapter. The main focus will be on QoS.

5.3.1 Fault management

The change in network topology in MANETs is high due to nodes moving around, nodes dieing or nodes going to "sleep" to save energy. This is not the case in wired networks, where the topology remains stable. When a link goes down in a wired network, this is treated as a fault. The problem then needs to be identified, isolated and corrected if possible. Because this is likely to happen very frequently in a MANET, this can not be treated as a fault in the same way. Links going down and new links being created will be normal happenings in MANETs, and do not have the need to be corrected. It is however important for the management node(s) to collect information of all these changes to keep an up to date overview of the network. This could be shown in a topology map with all the active nodes and the links between them. It can also be desirable to log all these changes and store information of each node in a database. Both due to the fact that nodes are likely to get reconnected several times during the lifetime of a MANET and due to the fact that the history of the network can give vital input to the development of managing such a network.

It is important that most of the network nodes are connected to the same network. Nodes that are about to loose network connectivity are therefore desirable to detect, and the manager must be alarmed. If nodes are just changing their point of connectivity, an alarm in the management node will not be desirable. Nodes may loose network connectivity both because of moving out of the network range, and because of loosing or saving battery power. In both cases it would be desirable to detect and alarm the manager before this is actually happening, in order to be able to avoid this to occur.

Because of the limited and variable bandwidth in MANETs, bottlenecks are likely to occur. This will again degrade the network performance considerably and make data transmissions for many nodes insufficient. It is therefore important to detect bottlenecks

when they occur and alarm the manager. Sending out new relay-nodes to take over some of the traffic may be a solution to the problem.

The task of fault management is only relevant during the lifetime of a MANET. What to be considered as a fault in a network is up to the network manager. Some faults are more important than others and not all type of faults are interesting to know about. As we have described above, faults in a wired network and a MANET will not be the same and have different priority.

Links going down may cause a network to get partitioned periodically or permanent. In these cases each partition needs to be managed autonomously. The partitions must also be able to merge back together and be managed as a single unit. The presence of co-existing networks is also a possibility. For instance, a police unit and a medical unit have their own network in the same physical area during a rescue operation. These networks may merge together or be managed independently and maybe just exchange some specific information.

5.3.2 Configuration management

If we use the scenario described in the introduction to this chapter, it is easy to understand that time is a critical factor. Rescue personnel participating in the operation needs equipment ready to use. It is therefore important that all equipment is configured right and working properly in their intended MANET. Of course, this configuration needs to be done in advance.

There might be different groups of people participating in a MANET, for instance policemen, firemen, doctors, nurses, network managers, etc. Each of these groups may have different needs for access to specific information and for the possibility to do administrator tasks. Such groups and their belonging network-rights needs to be configured before the network is put into operation, but this must also be possible to redefine during the networks lifetime. Also, new network users must be able to define on the way.

To be able to provide some sort of QoS, it is important to configure this in advance. The nodes functioning as routers must be configured with the necessary software and implement the policies on how network traffic should be treated. Because of MANETs mobile and unstable nature, guaranteed services will be hard to obtain. However, some sort of differentiated service could be an alternative to the best effort service, which is the default service. The configuration management can decide to give different type of traffic different priorities, and hence different treatment in the routers (described in chapter 3.4.2). It could be desirable to prioritise both different type of traffic and different type of groups. For instance, giving traffic originated from the network manager higher priority than traffic originated from a nurse, or giving real-time traffic higher priority than file transferring. Change of priority may also be desirable to configure during an operation.

The different network nodes need to have an IP-address. This address can be configured to be static, or the network nodes must be configured with a DHCP-client so that an IP-address can be received when the network is put into operation.

The task of configuration management is also important in order to do the other functional management areas. For instance fault management, where the nodes need to know when a critical level of battery power is reached, before an alarm can be sent out. Setting such different thresholds is therefore a task within configuration management and needs to be done in advance, but may also be reconfigured during an operation.

As we described above, obvious faults in a wired network is normal incidents in a MANET. A router dieing would be treated as a fault in a wired network, while in a MANET where this is likely to happen, this could just be treated as a configuration change. The management node registers what has happened and marks the node as dead. In [2] there has been proposed a table with configuration changes that is likely to happen in a MANET and their belonging actions to be done automatically at the network manager station.

Table 1 Network configuration changes [2]

Configuration Changes	*Action at Network Manager*
Node dies	Mark node as dead
Node is disconnected	Mark node as disconnect, await reconnection event
Node is powered off	Mark node as unavailable
Node powers back on	Mark node as available
New nodes join the network	Add entry for these nodes
Network gets partitioned	New manager started in each partition
Partitions merge	One manager takes over (choice determined by hardware and software capability, remaining power
A new co-existing network detected	Managers acknowledge each other's presence see if the networks can be merged
Node multiply managed	Allow this condition in some cases

5.3.3 Accounting management

MANETs are typically intended used in rescue- or military-operations where this type of management do not seem important at the moment. Reasons for this are that such networks are spontaneous and private. It makes no sense to bill the different users during for instance a rescue- or a military-operation.

5.3.4 Performance management

Because of the limited and varying capacity in a MANET, the possibility for bad performance now and then is much higher than in a wired network. If too much of the capacity in the network is in use, this will affect and degrade the overall performance in the network, and hence the task of providing QoS will be more difficult. Nodes are likely to be the relay-point for many simultaneous traffic-flows and bottlenecks will arise if these nodes capacity isn't high enough. It will therefore be important for the manager to monitor the traffic on each of the nodes and the links in the network, in order to discover parts of the network with poor performance.

The performance in the network influences the networks possibility to provide the requested QoS. Monitoring QoS parameters like, throughput, delay, and packet loss is therefore important to control how the network is performing during its lifetime. By setting thresholds on these variables, the manager can be alarmed when such thresholds are exceeded.

One part of the management task is to monitor and discover bad performance conditions in the network. Another task is to figure out what to do when for instance a congested node or link is discovered. If the discovered problem is continual, some efforts might be advantageous to be done. For instance, if a node is highly congested and in addition is an important gateway node for communication towards a hospital, a physical action like putting out a new gateway node to take over some of the traffic might be adequate. Another solution is to request for nodes to move, in order to change the topology and then achieve better network conditions. If the MANET is supporting some sort of QoS-level apart from the best effort, or if the network nodes are sending traffic viewed in the light of some decided policies, then it might be adequate to do some changes. For instance, giving some vital traffic high priority or decide that no transmission of video is allowed in order to reduce the amount of bandwidth.

Limitations according to the battery capacity, is a challenge in managing a MANET, which is not present in a wired network. Collecting information about the battery capacity is therefore necessary in order to avoid unwanted situations. Such situations can for instance be partitioning of a network or that a gateway- or management-node goes down. Avoiding this to happen, an administrator can physically give the node a new battery or delegate the task of being gateway/management to another node.

The task of performance management is just relevant during the lifetime of a MANET.

5.3.5 Security management

Security management is an important task both for fixed networks and for MANETs. However, security issues in MANETs are even more difficult because of the fact that everyone with the right equipment can listen to the traffic. MANETs are in addition

intended used in hostile environments and environments with a lot of sensitive information flowing between the nodes, making the task of security even more important.

To achieve a secure network, it is important that the users of the network are authenticated. Access control, including users to type a username and a password, is a minimum security demand. The manager therefore needs to configure the access control in advance. This includes defining the legal users and their belonging rights in the network. The possibility to add new users is also a need during an operation. If strangers are trying to access the network, notification at the management node could be desirable. Information sent in a MANET needs to be encrypted, in order to avoid this information to be tapped by intruders. If the receiver of the encrypted message shall be able to read the message, it needs to be decrypted. A safe solution is to equip all valid network users with a common key in advance.

Security matters also have an influence on the QoS-providing. Unwanted persons might be a great risk to the network, not only because sensitive information might go astray, but also the fact that the network performance can be sabotaged.

5.4 Different importance of management

The task of management will not be identical to all types of MANETs. MANETs differ in the environment where they are intended used and they differ in architecture and size. To propose one standard for how management in a MANET should be defined is therefore inappropriate.

In MANETs where the information sent is extremely confidential, security management will be of high importance. In other networks where demands to low delay are more important, the focus will be on performance management. It is important that the utility value of the management task is in relation to the network resources it occupies. Another factor that can affect the importance of management is the size of the network. More nodes lead to higher traffic, more movements and therefore bigger possibilities for something to go wrong. Management is important to prevent, discover and solve potential problems that might occur.

5.5 Challenges in managing a MANET

Management of a MANET is a very challenging task [2, 6, 9]. Several different characteristics contribute to underline this statement, and they are as follows.

5.5.1 Dynamic topologies

Nodes in a MANET are wireless and they are free to move in an arbitrarily manner. They may also be located in or on cars, ships, people etc. Because of this, the topology is likely to change frequently. Changes in topology is caused by; a new node or subnetwork getting added/deleted, failure and limited survivability of a node or a link, and nodes changing their point of connectivity.

It is important that the manager gets an updated presentation of the network topology. Monitoring a networks topology is an important task in all network management systems. The more the topology is changing, the more the signalling overhead increases, due to present an updated topology in the management node. Because of restricted bandwidth and low battery capacity, it is important to minimize this signalling overhead.

Wired networks have much less topology changes, reducing the signalling overhead. Updating the topology is therefore a much more challenging task in a MANET than in a wired network, due to minimizing the signalling overhead.

5.5.1.1 Network partitioning

Network partitioning occurs when a subnetwork loose the connection with the rest of the network because of too long distance. Dynamic topology and nodes running on batteries can result in frequent partitioning. When a network gets partitioned, the subnetwork(s) need to discover this and work autonomously.

It is important that the manager is alerted when a partitioning occur, because this makes possible essential information to be sent between a node in a subnetwork and a node in the main network impossible.

5.5.1.2 Network merging

Network merging is the opposite of network partitioning. It is important that the network recognize this and work as a single unit. The signalling overhead in this process must also be kept to a minimum.

5.5.2 Low bandwidth and variable link capacity

The bandwidth in MANETs is in average much lower than in wired networks. Because of this, congestion is more likely to occur and the network performance degrades. It is much harder to offer QoS in bandwidth restricted networks, where congestion is closer to normal than it is an exception.

In addition to the low bandwidth, different other factors like, multiple access, fading, noise and interference, limits the actual throughput in the network. These factors are not constant, leading to a variable link capacity. Multimedia applications with increased demands to the network will therefore be challenging to satisfy.

The signalling overhead used by the management protocol is therefore also important to limit, because of the restricted bandwidth in MANETs.

5.5.2.1 Bottlenecks

A MANET may consist of hundreds of nodes, and the topology is very unpredictable. A node is therefore likely to be a relay-node for many traffic flows. If the capacity in that relay-node isn't high enough, congestion will occur. A bottleneck like this is a problem for all networks, but especially in MANETs where bottlenecks are more likely to occur, due to low bandwidth and low storage and processing capacity.

The problem of bottlenecks in a MANET is an important and challenging task for the management of such networks. Bottlenecks are especially a treat to provide QoS, because latency and packet loss will increase.

5.5.3 Limited resources

Most of the nodes in a MANET run on batteries. It is therefore important to limit the network management overhead to save energy. Energy is used by the nodes when packets are transmitted, received or processed. Also resources like storage and processing capabilities are limited due to portable and light nodes.

A node running out of battery and a node going to sleep to save energy, changes the topology in the network, and makes the management task more challenging.

5.5.4 Heterogeneity

Nodes in a MANET have very different complexity, varying from sensors and PDA's, to fully functional computers. All nodes in the network are therefore not equally suited to serve as head-management nodes. For instance, a sensor will probably contribute minimal

to the management task, while computers with the best available resources will have the emphasis of the management tasks.

5.5.5 Security

MANETs are often set up in environments where the needs for security are crucial, like in military operations and rescue operations with a lot of sensitive data flowing between the nodes. These operations will also be more frequently exposed to security attacks like eavesdropping, spoofing, denial-of-service, destruction and penetration. It is therefore very important that the management protocol to be used have solutions to the mentioned treats, including authentication, encryption etc.

5.5.6 Multiple roles

In MANET, most nodes are expected to play different roles. A node might be a router and forward packets, and at the same time be a source or destination for different application flows. This leads to higher complexity, having in mind the high mobility in a MANET.

5.5.7 Avoid unnecessary topology changes

Because of the variable link quality, due to fading, noise, interference etc, links may go down periodically. This is not a change in the physical topology, and therefore topology updates in the management node are unnecessary. Also, nodes going to sleep to save energy are an example of the same type. This is interesting due to save the network from unnecessary overhead.

5.5.8 Providing QoS

All the mentioned characteristics of a MANET, makes the providing of QoS difficult. The low bandwidth and the fact that many nodes will compete for this limited resource, will cause difficulties in satisfying all users and their demands to QoS. The dynamic nature of a MANET and the variable link quality makes a MANET very unpredictable, and it is therefore hard to give any guaranteed service to a user. Making the network running effectively will therefore be an important task for the management.

5.6 Different management architectures in MANETs

MANETs differ in the number of nodes connected and situations they are meant to operate in. Because of this, different management architectures will be suited for different MANETs, which will be discussed in this chapter based on the architectures described in chapter 4.3.

The centralized architecture is the simplest one to implement, and suites for small and centred networks. This type of architecture has some drawbacks, and these drawbacks increases with the growth of the network. To control the network, one single management node is responsible for collecting data from all the other nodes in the network. This leads to a lot of message overhead, which is important to minimize due to limited resources. A single management node will also be vulnerable for faults in the network. If this management node experiences a failure, the rest of the network is left without any management. Also, if the network gets partitioned, the subnetworks without connection to the management node will be left without this functionality. The advantage with a centralized architecture is that there is only one management node, making security attacks more difficult.

The distributed management architecture consists of more than one management node. These management nodes communicate with each other in a peer-to-peer manner, making the network less vulnerable if a management node fails. Each of the management nodes controls a subnetwork, and therefore this type of architecture is suited for bigger networks than the centralized architecture. The message overhead will also be reduced due to less average hops from a node to its manager, which is an important factor in a MANET.

The hierarchical management architecture is suited for large scalable networks. It has different levels of management nodes, with one head management node on top of the hierarchy. This can be compared with a tree structure, with the head management node as the root. Each management node in the hierarchy has its own domain. Messages between the managers are only sent along branches in the tree-structure. An intermediate node collects information about its domain, and decides if this information is needed to be sent to an upper level in the hierarchy. Messages can also go in the other direction, with the head manager sending information to all or specific management nodes. This type of architecture is very favourable with a view to minimize the management overhead, which are important in MANETs.

The Policy-based architecture is also intended used in large scalable networks. On top of this architecture there is a PMT where policies are defined, and then stored in a Policy Repository database. A PDP will then deliver these policies to the PEPs on demand, or when special events occur. This will reduce the management traffic between the manager node (PMT) and the nodes being managed, because of the nodes autonomous nature. It would be desirable to have several PMTs in a MANET in case of networks splits etc. to make the architecture less vulnerable. We will in chapter 6.5 present a Policy-based framework designed for use in MANETs.

6 Management solutions

6.1 Overview

In the previous chapter we discussed needs for management and different management challenges to overcome in a MANET. In this chapter we will present different management solutions for use in MANETs, and discuss these solutions based on the theoretical challenges and state how suited they are for use in such networks. This is according to activity 3 in the task description (chapter 1.2)

We will first present the SNMP solution in chapter 6.2. This is the most used management method in wired networks today. In chapter 6.3 we describe ANMP, and in chapter 6.4 we describe the Guerrilla management architecture. These two management approaches are experimental solutions for MANETs. In chapter 6.5 we present the Policy-based framework for MANETs, and finally in chapter 6.6 we will do an overall discussion of the presented solutions.

6.2 SNMP

6.2.1 Introduction

SNMP was introduced in 1988, and has since developed to be the main standard in network management solutions. Internet Architecture Board (IAB) has accepted and recognized SNMP as a standard protocol. RFC 1157 [3] describes the agent/manager model used in SNMP. An agent is software capable of answering valid queries from an SNMP manager about information defined in the Management Information Base (MIB). SNMP is an application layer protocol and uses UDP and IP for data transmission. One of the advantages of SNMP is its simplicity. It is easy to implement and use in multi-vendor networks.

6.2.2 Architecture

The main components of SNMP [26, 27] are manager and agents. The manager is the device that supervises the network, stores the information collected etc, while the agents are the nodes being managed and they are serving as an interface between the specific agent node and the manager. All the information collected at each node is stored in a virtual database called MIB. SNMP handles the communication between the agents and the manager when transmission of MIB information is needed (Figure 12).

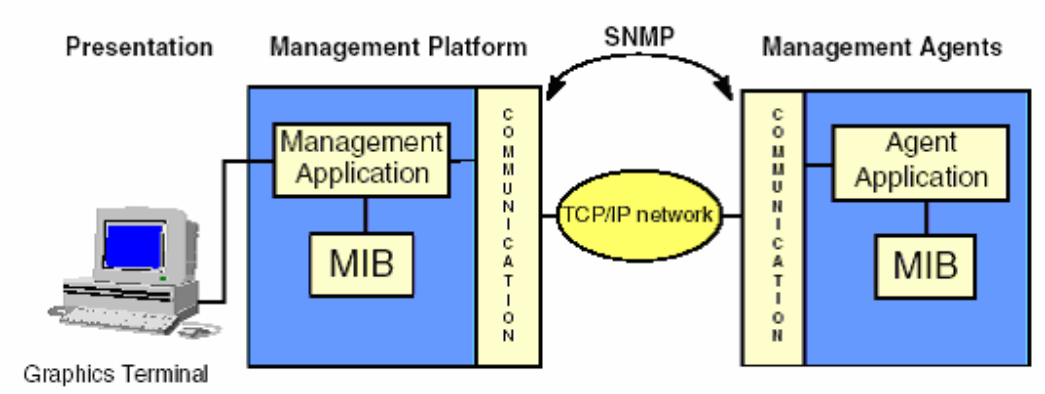

Figure 12 SNMP overview

There is a need to organize the network in different structures like an administrative structure, an information structure and a naming structure. If the agent/manager relationship shall work, it will be necessary to have a structure for logical exchange of available objects. Structure of Management Information (SMI) is a framework that organises names and describes the different objects. It states that there must be unique

name, syntax, and encoding. The ISO and the CCITT have suggested organising all the MIB- objects in a global naming-tree and then assign an identifier to each of the objects.

The objects we want to manage in the naming-tree are located on leaf nodes and they are labelled with a series of integers and a short text description. This series of integers are organized in a hierarchical way, with the root integer first and then an integer for each of the new branch in the naming-tree. This series of integers are called the Object Identifier (OID).

The syntax is written in a data-type definition language called Abstract Syntax Notation 1 (ASN.1). For a while it was normal to define data communication specification formats bit by bit. CCITT proposed a solution where the formats where defined on a high level language tool and let a compiler make the translation to machine codes.

For the encoding on the physical layer between the sending- and receiving node, the Basic Encoding Rules (BER) is used.

6.2.3 MIB

Enormous variation of devices in the internet today delivers different ability to present their device specific information. Status from a router port or information about routing tables might be desirable for the manager to know about, and it must therefore exist a way to get this information. A logical database called Management Information Base (MIB) handles this problem. Managers are dependent on the agents on the devices to deliver the needed data to the MIB. From the period of 1988 to 1991 there was a great progress in the development of the MIB, and this is today in the MIB 2 version. Over the resent years this has proved to handle the IP management in a satisfactory way.

When defining MIBs for use in the Internet management framework, it is important to keep them simple and light. The resources on an agent node, like for instance a sensor or a PDA, is limited and it would therefore be of interest to minimize the MIB objects. It would not be an adequate use of resources if a node uses all its computed resources on handling management queries. To lighten the work load on agents, management applications running on the manager node often calculates information from MIB objects to minimize the polling.

In the architecture of the MIB-tree, it has been focused on the ability to easily extend the MIB with new parameters. This makes it easy for private vendors to make specific MIB modules to their products.

The different variables in the MIB are gathered in groups based on how they function. Each group have an identifier in the MIB tree, and the groups are again divides into objects. The needs for all the functions that the different groups deliver, differs from node to node. For instance, a bridge would not have the need for the UDP-group, and such a device would perform better by disabling such unnecessary functions.

6.2.3.1 Internet management MIB

The MIB objects in the Internet management MIB are placed in groups dependent on the functions they deliver. The Internet management MIB consists of 12 sub-groups. We will here present 11 of them (figure 13), and give a small presentation of what they provide. Examples of object information in the sub-groups are tables, like for instance routing table, counters for different frames, both in octets and number of frames, and static values, like for instance what interface(s) a node contains.

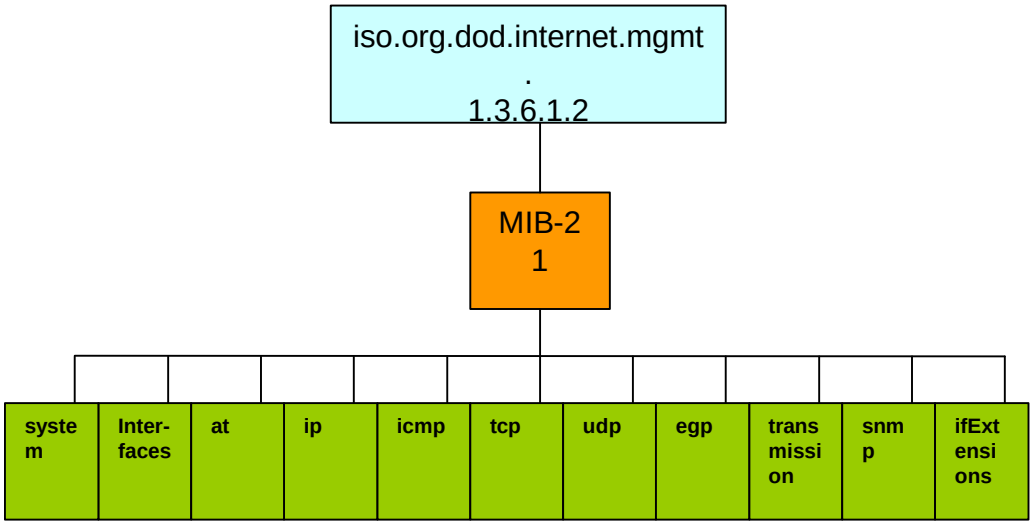

Figure 13 MIB2 tree

6.2.3.1.1 System group

This group is essential for all nodes to deliver SNMP functionality, but all the variables do not need to have a value. It delivers system information of a specific node, like the Operative System (OS), the SNMP version it is running etc. It also delivers information on where it is located and who to contact when problems occur.

6.2.3.1.2 Interface group

The different nodes communicate with each other through network interfaces. MIB 2 support over 100 different interfaces today, and there are more to come. The interface group is mandatory to every node, and it tells for instance what type of interface and speed that is running in a node, the status of the interface, traffic statistics and error counts etc. There are different solutions for wireless technologies under development, where you for instance could collect information about signal strength on the sending- and receiving signal.

6.2.3.1.3 Address translation (at) group

The function to the address translation group is to map MAC addresses with IP addresses. This group seems to be out of date, and it was described in MIB 2 to be compatible with the first MIB spec.

6.2.3.1.4 Internet protocol (ip) group

The intention of this group is to provide information about IP operations, the routing table and the mapping between physical and network addresses. The group consists of several individual objects and it has three tables collecting bulks of information.

6.2.3.1.5 Internet control message protocol (icmp) group

This group is used to report messages from the ICMP module, and consists mainly of counters for errors and incoming and outgoing ICMP messages on the nodes.

6.2.3.1.6 Transport control protocol (tcp) group

This group contains information on incoming, outgoing and error statistics of TCP traffic, and gives information on the maximum number of TCP connection allowed. The connection table in the TCP group allows the manager to see the active TCP connections.

6.2.3.1.7 User datagram protocol (udp) group

Since this is a connection-less protocol, there are not many entries to collect here. But the group delivers statistics on incoming and outgoing traffic, plus error counts.

6.2.3.1.8 Exterior gateway protocol (egp) group

The EGP group contains information about incoming and outgoing traffic on gateway nodes.

6.2.3.1.9 Transmission group

This group contains different transmission technologies and traffic variables for each technology.

6.2.3.1.10 SNMP group

The SNMP group contains information about the SNMP objects regarding traffic flow and errors. In case of large amount of polling, this group can give us a good idea of the resources SNMP absorb in the network.

6.2.3.1.11 ifExtentions group

This group is in "family" with the interface group, but there is added some new functions that is not supported in the interface group. It separates broadcast and multicast traffic, and it contains a list on which physical addresses the interface will absorb traffic.

6.2.4 SNMP protocol

It is important to separate the MIB database from the SNMP protocol. While the collection and storing of information is handled by the MIB, the transport of information is handled by the SNMP protocol. The protocol is necessary in order to transport management information, either when the manager request information from the agents or when agents on their own initiative send information to the manager when special events occur. The process of collecting information from the network nodes, are done by polling, and management software are designed to do this periodically. There are three types of messages that the manager may send to an agent, namely Get, Get Next and Set. The agents have two possibilities, either a Response to the management queries or a Trap-message (Figure 14).

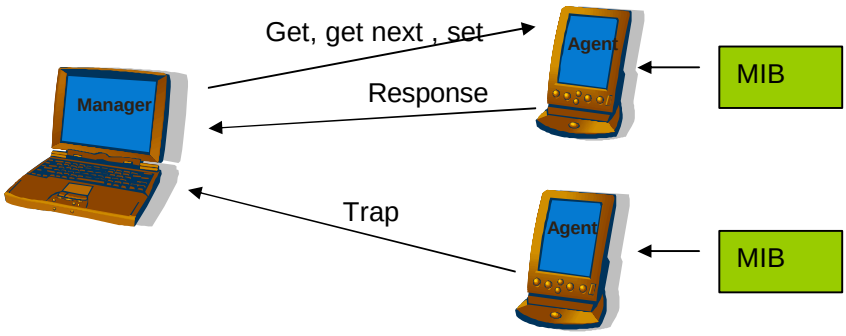

Figure 14 UDP traffic

The SNMP protocol was designed to manage Internet nodes, and the chosen Internet protocol was TCP/IP. The decision to use UDP instead of TCP was because of its simplicity. TCP is connection oriented and use a lot of memory and processor resources, which is not the case for UDP. It was then natural for SNMP to choose UDP, since also this transport protocol runs over IP.

6.2.5 SNMP PDU

The SNMP v1 message format consists of two parts, a message header and a Protocol Data Unit (PDU) (figure 15). The message header contains two fields, a version number describing the SNMP version, and a community name used in connection with authentication in a group.

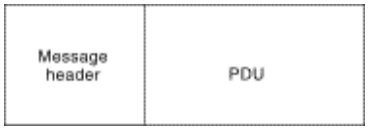

Figure 15 SNMP PDU frame

The PDU includes a set of commands and operands that indicates the state of the object involved in the transaction. The length of the SNMP PDU is variable and may contain two different frame types, one for Get, GetNext, Response and Set, and one for Trap.

As for the first PDU (figure 16) there is first a PDU type field describing the type of PDU transmitted. Request ID is to associate query and response messages. Error status is only used by the response messages and indicates errors. Error index is only used by responding messages and associate an error with a specific instance of an object. The Variable bindings field serves as the data field or payload.

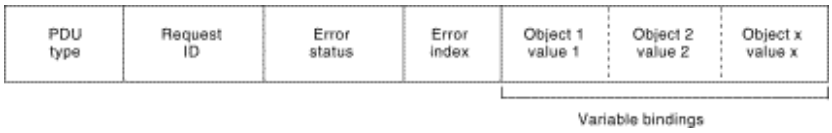

Figure 16 SNMP PDU frame for Get, GetNext, Response and Set messages

As for the event based Trap messages, the frame format is slightly different (figure 17). It starts with an Enterprise field that identifies the object that generated the Trap. Agent address gives the address of the managed object generating the trap. Generic trap type indicates one set of generic trap codes. Specific trap codes indicate one set of specific trap codes. Time stamp gives the time since the last trap messages was generated. Variable bindings field serves as data field or payload.

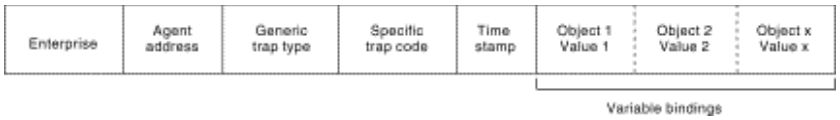

Figure 17 SNMP PDU trap message

Figure (18) shows the possible information flow between a manager and an agent in a management system using SNMP v1.

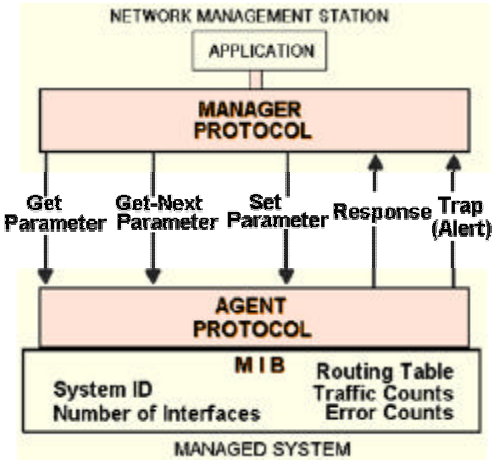

Figure 18 SNMP PDU message stream

Get Request messages is used to query MIB variables on the agents.

Get-Next Request is like the Get Request, accept from that it queries a sequence of MIB values. This is useful when for instance collecting a routing table.

Set Request describes an action to perform, and updates one or several MIB values.

The Get Response message responds to the Get-Request, Get-Next Request and Set request by delivering the queried MIB object.

Trap messages can rapport on critical occurrences in the MIB values on an agent. The trap messages are received on the manager without the need to poll this information. There are defined six traps; SNMP v1 cold start - sender is reinstalling and changes in the configuration might occur, warm start - reinstalling but no changes in the configuration, linkDown - failure in a link, linkUp - a link is up/restored, authenticationFailure - a protocol message is not authenticated, egpNighbourLoss - EGP neighbour is down, and enterpriseSpecific - a form to define vendor specific trap messages.

6.2.6 SNMP v2

The goal of upgrading SNMP to SNMP v2 was to add some security, but it mainly ended up with adding a few error codes and some more efficient ways to retrieve data. The second version never made it to be an IETF standard. A few vendors added its functions into their management applications, but it never really got accepted.

SNMP v1 did not deliver any good solution for security issues like authentication of the source of the message, protection of messages against exposure and having access control on the MIB databases. These drawbacks were tried improved in SNMP v2.

Two protocol functions were also proposed. The first one added the ability to effectively send large amount of MIB data, and the other one added the possibility for manager to manager communication when critical events occur in the network.

6.2.7 SNMP v3

In this version there were implemented more powerful security functions. These were features including access control, authentication and privacy of management information, which were not solved in the second version. In version three there were included a User-based Security Model (USM) that provides message level security like avoiding modification of information, masquerade, message stream modification and disclosure. Another security function added was the View-based Access Control model (VACM). This is mainly to provide access control to the MIB. Two drawbacks with this version are the complexity and the lack of backward compatibility.

6.2.8 RMON

Due to the mobile nature of MANETs, a direct polling of agents increase in difficulty as the network grow large, and also overhead caused be this polling should be reduced. In large networks we may presume that a network is likely to divide into two or more parts from time to time. It would therefore be desirable to collect information several places in the network, and rather send the collected information to the manager on request. Remote network MONitoring (RMON) is a MIB designed to collect real-time data and historical MAC-layer statistics. RMON MIB defines basic methods to collect management information on LANs, and a RMON agent would be placed on each subnet in order to listen to the medium and store/capture the information it is defined to do. It also contains an alarm and an event sub group where thresholds can be set in order to notify on severe changes in the network. Using RMON to analyse and monitor network traffic from a central location may provide the manager with information about critical occurrences in the network before they result in a crash.

A RMON-setup normally has a network management station and a remote device or the RMON agent. The manager station sends SNMP commands to request information on the RMON agents. The RMON agents will then respond to these queries and send the desired information back to the management station, which in turn will analyse the data and display it graphically. As shown in figure 19, the overall manager may lose contact with a part of the network. It would then be convenient to have an RMON agent on the separated part of the network in order to collect information from the separated nodes. Information about the traffic-data and the active nodes in the divided group could be of

interest. The last on is very useful for the manager if/when the isolated group of nodes comes in contact with the manager again.

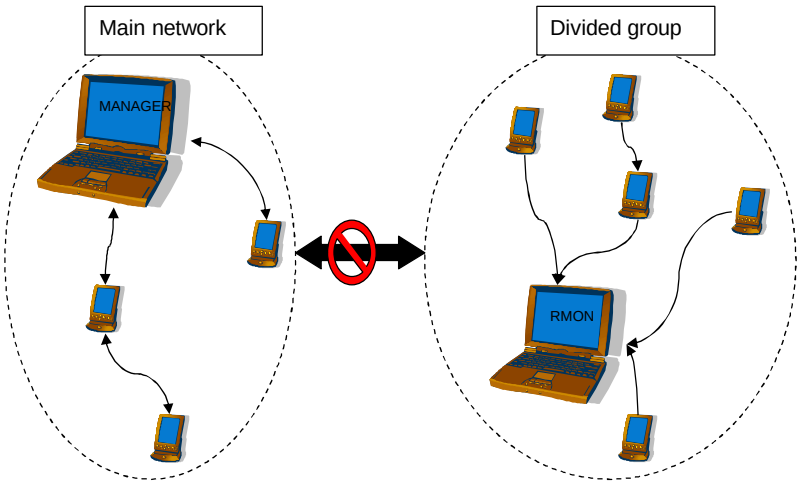

Figure 19 RMON in MANET

6.2.9 Discussion

SNMP is a protocol originally invented for wired networks and is the most widespread management solution today. The most common setup for SNMP is a manager that has responsibility for a set of agents. This might work well with a few agents, but as the network grows this becomes a problem due to the overhead created when the manager polls the agents. This would be critical for the limited and unstable bandwidth in a MANET. Trap messages are a function to reduce this problem. If an agent discovers a serious condition, it can tell the manger without being polled. This prevents the manager from polling all the network nodes periodically for information that are not useful. On the other side, too much use of trap messages when the network experience bad conditions, might just harm the network even more. An administrator does not need trap messages when he/she already know that the network performance is bad.

Another problem with a centralized management structure is the one that occur, when the network splits and the manger loose contact with some of the nodes. To attack this problem SNMP provides a decentralized method with the help of RMON MIB. This is an extension to the MIB II that is standard for most SNMP solutions. The RMON agents prevent the overall manger from polling all the agents directly, which reduces the management overhead considerably. It also makes the network able to collect information when the network splits, which can be delivered to the top manager when the network merges back together again. One drawback with the RMON MIB for use in MANETS is that it is designed for working in wired LANs and therefore listens to the whole subnet and not just parts of it. The use of RMON in MANETs will therefore be insufficient.

Because SNMP has been available for such a long time, there are a variety of solutions available, ranging from open source projects like UCD–SNMP and commercial solutions from Hewlet Packard and IBM. These applications graph information, manage polling of object and make you able to browse your MIB. Some advanced software make you able to compose your own MIB, but most network related devices like routers and switches comes with vendor MIBs. Today there is not much MIB support for wireless technologies available, like for instance 802.11b. This makes the collection of information regarding antenna strength, packet loss etc. very difficult.

6.3 ANMP

6.3.1 Introduction

Ad hoc Network management protocol (ANMP) [2] is a management method designed for MANETs. It shares a lot of the same characteristics as SNMP, but has some extensions regarding the architecture, the MIB and the task of providing security. These are areas where they mean SNMP is insufficient for MANETs.

6.3.2 Architecture

ANMP has a three level hierarchical architecture (Figure 20). The bottom level consists of agents, and these agents are collected in groups called clusters. In each cluster there will be a cluster head that manage the agents. At the top level there is a network manager that manages the cluster heads. The anatomy of the cluster heads is dynamic. This is because of the mobile nature of MANETs, and nodes may move around and change clusters and cluster heads might move or fall out. ANMP presents to types of algorithms for clustering; the first one is based on the graph topology and the second one uses a global positioning system to form clusters based on the tightness of nodes.

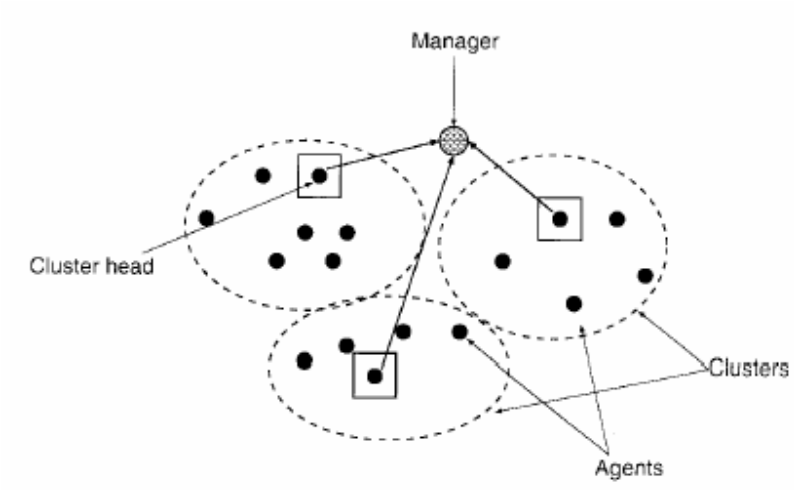

Figure 20 ANMP architecture [2]

The agents in ANMP collects the information locally based on what the manager wants, puts it in its MIB and sends it to the cluster head above. Then the cluster head filters the information and systemise it. The cluster heads have tables with information about each of the agents under its control. At last, the cluster heads send this information to the overall manager. ANMP uses the SNMP-Packet Data Unit (PDU) for the exchange of information between the agents, cluster heads and the network manager. One difference

in ANMP compared to SNMP is that there is no retransmission of SNMP PDU, because objects are updated periodically. However, the manager has the ability to overrule this function and specify requests for lost packets.

6.3.3 anmpMIB

Agents uses an extension to the SNMP MIB II called anmpMIB (figure 21), that contains four sub groups; power usage, topology maintenance, agent information and LACM. The power usage group keeps information about the energy consumption in the nodes. One special object is the powerBatteryDrainFunction that states the power of the battery and the remaining lifetime of the battery. The topologyMaintance group contains information about the topology in the network. It has entries for the protocols handling clustering, and today there are two protocols added here, hence graph-based clustering and geographical clustering. The agentInformation group is used to store information from the agents on the clusterheads and the manager, but if needed it is easy to add new subgroups. All the subgroups in agentInformation are used to log data except from the alarm- and event group.

The last group, LACM, contains information about classification and security clearance of the mobile agents. This is important since the manager and the clusterheads need higher clearance than the agents.

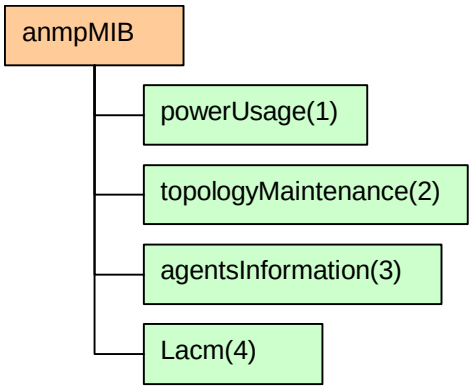

Figure 21 anmpMIB

6.3.4 Security

ANMP contains the same unicast security as in SNMPv3, but in addition to this it also support secure multicast and the military security model. If the manager is about to collect sensitive information from the nodes in a MANET, the SNMPv3 has to send single messages to each node to obtain the desired security. In ANMP it is possible to send multicast secure messages to all the nodes, which again will reduce the overall overhead.

In the military security model it is set clearance restrictions on nodes, and clearance classification on data. By doing this, a cluster head can not access information beyond its level, but a clearance level equal to or lover would be possible to access.

6.3.5 Discussion

The design goal of ANMP was to make a lightweight protocol that was compatible with SNMP. The main reason for this is that the most widespread management protocol today is the SNMP protocol.

The similarity with SNMP is salient in many ways. The PDU used by ANMP has the same PDU structure as the one SNMP uses. ANMP also uses the UDP protocol to exchange messages between the manager and the agents. ANMP does however not retransmit lost data, which is in contrast to SNMP. This is because the information is updated periodically anyway. It is however possible for the top manager in ANMP to request for lost data.

ANMP is based on the RMON-technology, and the cluster heads therefore collect information in the same way as RMON in SNMP. The biggest difference is the clustering algorithms implemented in ANMP. While the RMON-agents control different subnets, the cluster heads in ANMP control a group of nodes based on a clustering algorithm. The clusters are also very dynamic and the task of being a cluster head may change.

The ability to monitor the battery capacity and the draining speed makes it easy to estimate remaining life time of the batteries. This is very interesting for the manager to know about in a MANET, and it is an improvement compared to SNMP. The manager has then the possibility to send out a new battery to a node before it is dieing due to lack of battery power.

6.4 Guerrilla management architecture

6.4.1 Introduction

The main goal for Guerrilla [4] is to solve the unpredictable behaviours of MANETs. The main difference from other manager to agent solutions is that Guerrilla has a Client/Agency (figure 23) solution instead of the usual Manager/Agent model. This solution has the ability to divide management tasks to different nodes in the network regarding to how they are suited.

6.4.2 Architecture

The ability to perform different management tasks differs from node to node in heterogeneous MANETs. Small devices like PDA's or sensors might just manage to respond simple management queries from the managers, while powerful laptops often can execute advanced management tasks. According to the resources available in each node and the network dynamics, the Guerrilla architecture classifies nodes into three levels (figure 22).

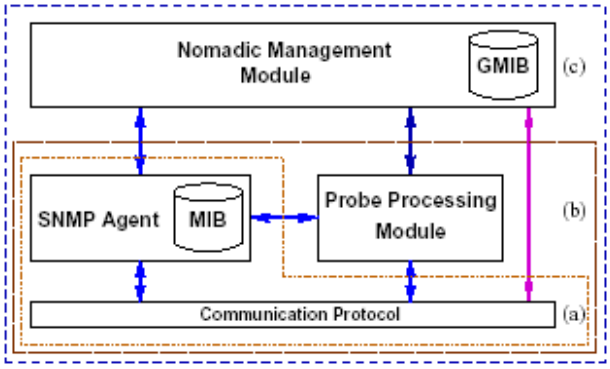

Figure 22 Guerrilla levels

On the bottom level there are nodes that are only capable of executing an SNMP agent in order to serve remote access to local management information. Level two of the architecture adds a Probe Processing Module (PPM) on top of the SNMP agent, and is implemented on nodes with enough resources. The PPM is a simple execution environment that is capable of executing active incoming probes. This makes it capable of query SNMP agents in order to process MIB information and poll remote SNMP agents. The probes encapsulate management information and send it hop-by-hop to its destination node. The top level in the Guerrilla architecture adds a Nomadic Management Module (NMM) on top of the other two levels. To obtain this role, the actual node must have enough power and processing resources. This module has different tasks like maintaining management information and states, communicating with other nomadic

managers, spread active probes to other manager nodes in its own domain, and migrate or spawn other nomadic managers according to network dynamics.

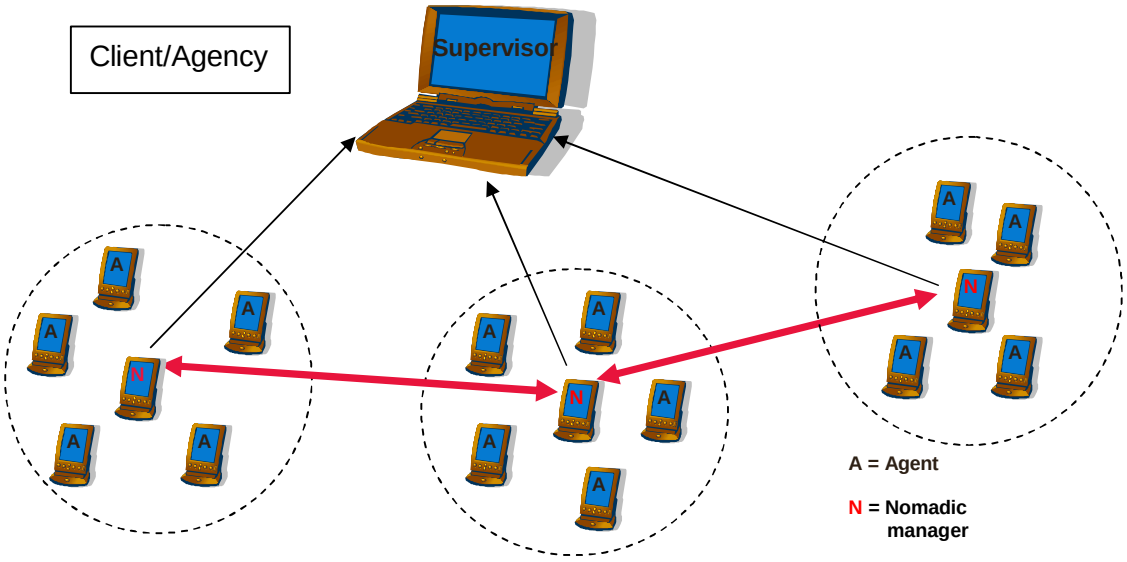

Figure 23 Guerrilla architecture

An important task in a network is the one of gathering information. In Guerrilla there will be a need to collect information from agents, and exchange this information with other nomadic managers. To solve the task with information exchange between the nodes in the network, Guerrilla introduces a small packet type called a probe. The probes are able to exchange more than raw data. It may also process the information in a node, do filtering, and forwarding a collection of useful information. Guerrilla introduces to types of probes, namely monitoring probes and task-specific probes. Monitoring probes is used when a nomadic manager wants to obtain network information from nearby nodes. The probes duplicate in order to cover the area specified by the nomadic manager, and they collect and send back network information to their manager. Task-specific probes perform specific operations, or collect special application information. The probes could be designed to collect custom information like topology data.

Nomadic Management Module (NMM) runs on top of a virtual machine called the Execution Environment for Nomadic Manager (EENM). It simplifies intra-domain communication as well as coordination with the modules outside the NMM, as the probe processing module.

All the management information obtained in the network, are stored in a structure like the SNMP MIB. The difference from the SNMP MIB, is that the information is gathered with probes. The collection of information is called Guerrilla Management Information Base (GMIB). GMIB can be accessed by NMM and SNMP agents as a branch in the SNMP MIB.

6.4.3 Discussion

One main feature of the Guerrilla management architecture is the distribution of management tasks among nodes in the network. This may be important in a MANET, like in a rescue scenario where the different participants have nodes ranging from PDA's to powerful laptops. Nodes with high performance capacity may take the burden of the heavy management tasks, while "light" nodes may just serve basic SNMP functions. The autonomous function in this management method makes it possible for nodes to delegate management responsibility to other nodes, when for instance a node is running low on power.

Another useful advance of this distributed management is that nodes with probe executable ability may gather information from its nearby nodes, and then send this collected information to its nomadic manager. This makes the waste of bandwidth minimal. And in case of a network split, the information collected will not be wasted. Nomadic managers have also the ability to exchange information among each other, in order to update and duplicate/spread information in case of network splits etc, resulting in a more flexible management.

Guerrilla is as mentioned just an architecture intended used in MANETs. It is made in a very general manner, which makes it very suitable for many purposes. It can easily be implemented in most networks, and SNMP would be a natural alternative together with the Guerrilla architecture.

The Guerrilla management method needs to be further developed before it can be a commercial solution, which makes it an unsuitable alternative for management in MANETs at the moment. It has however a great potential to be a good management solution.

6.5 Policy-based management

6.5.1 Introduction

The ability for a manager to perform some control over the network in order to make it more efficient, is desired in dynamic networks like MANETs. This is often done with the help of QoS functions. A Policy-based framework [6, 7] approach for MANETs is made in an attempt to solve these issues.

6.5.2 Policy-based transport

For the transport of policies, it is possible to use several protocols. There is however made a protocol called Common Open Policy Server (COPS) that is designed to transport policies. An extension of this protocol is made for PRovisioning, namely COPS-PR. COPS is a client-server protocol that handles the communication between policy clients and remote policy servers. There are two different control models, the outsourcing model (figure 24) and the provisioning model (figure 25). COPS supports the outsourcing model, while COPS-PR supports both of them.

The outsourcing model has a 1:1 relation between the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP). When an event occur that needs a new policy, the PEP will send a request (REQ) to the PDP, which in return will make a decision and send a decision message (DEC) back.

The provisioning model has a m:n relation between the PDP and the PEP. The PDP reacts to events in the network and distributes policies to the PEP. The PEP will then make its own decisions based on the policies obtained form the PDP.

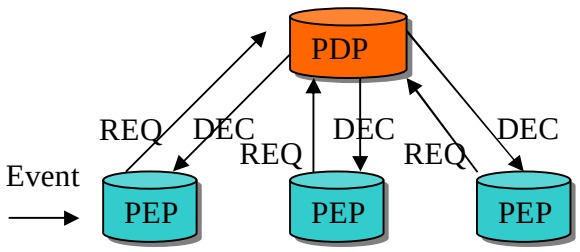

Figure 24 Outsourcing model

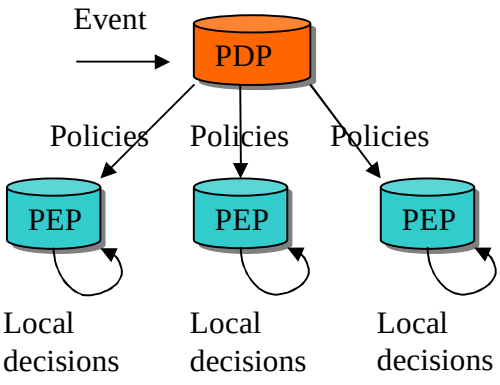

Figure 25 Provisioning model

6.5.3 Policy-based Framework for Wireless Ad Hoc Networks

The Policy-based management framework [6, 7] presented in this chapter, tries to give a deeper insight of Policy-based management for MANETs. It includes components that it consists of and how they function together. Some off the components presented is exclusive for MANETs, while others are just as good for regular wired networks.

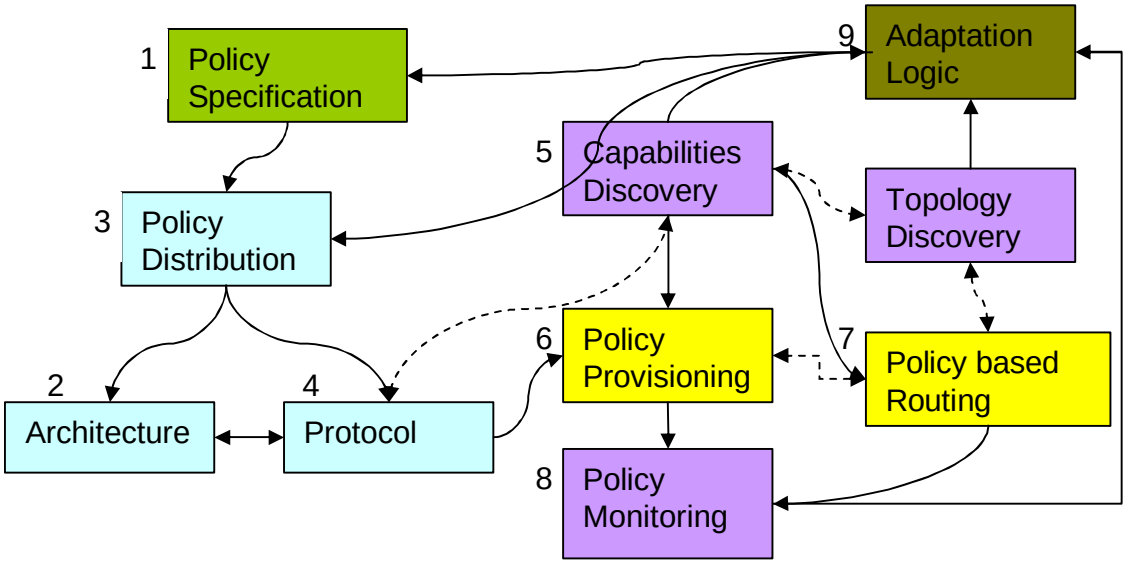

Figure 26 Policy-based framework for MANETs

Policy specification (1)

It is at this level that the administrator interacts with the PMT, and defines all policies. The overall network goals are mapped together with network policies. The high-level goals are specified by the administrator and are normally static, while low-level policies are often more dynamic, due to the devices they are going to interact on. The unstable nature of MANETs, often forces the manager to do high-level policies in a dynamic way.

Architecture/Clustering (2)

The chosen architecture is fatal for how the bandwidth constrains is handled in a MANET. For the Policy-based management, there is presented an outsourcing model and a provisioning model that are both good for handling special events. A combination of these to models would be suited for MANETs.

As mentioned in the ANMP chapter, clustering nodes in groups and form networks with interconnection of clusters is a good approach in order to ease the task of management, efficient routing, support routing etc. Each policy server forms a cluster together with its surrounding nodes. Policy clients within k hops from a server will get their policies from that policy server. This is done to restrict the number of hops between the server and the clients. A problem with this approach occurs when nodes exceed k hops away from a

policy server. This might happen because there are not enough policy servers to serve all the nodes, or the nodes may move out of a k-hop cluster. To approach this problem, Dynamic Service Redundancy (DynaSeR) is used. When a node moves out of its cluster, the server will collect topology information to check if the node is within k hops away from other servers. If this is true, it will redirect the client to the new server, and the client will be managed by this new policy server. When a node is more than k hops away from all existing policy servers, the server will delegate a network node to handle the delivering of policies to this node. This will add another control layer beneath the policy server, resulting in a hierarchical control architecture.

Policy distribution/Protocols (3, 4)

An effective and reliable mechanism for distribution of policies is an important part of the architecture. It exist several approaches for distribution of policies in a Policy-based network, like COPS, SNMP and Lightweight Directory Access Protocol (LDAP). It is important that the method/protocol used is efficient, lightweight and deliver a robust mechanism for QoS provisioning in a MANET. COPS and COPS-PR seems to be a good choice for QoS provisioning and management. Distinctive advances with COPS-PR is its event-driven control, resulting in no queries from the policy clients to the policy server Support for fault tolerance and security, and the use of TCP gives a reliable packet-transport. SNMP could function as a co-existing protocol for gathering network information, while COPS-PR could function in order to distribute policies.

Capabilities Discovery (5)

The policies in the network are being translated into device specific configurations in order to dictate the use of network resources. To be able to do this, the framework must have knowledge of the status of the network like the bandwidth, device capabilities, if devices are active or not etc. Such criteria are even more important in a MANET, based on their unstable nature.

Policy Provisioning (6)

After the policies have been distributed to the devices in the network, the policy provisioning phase will install and implement the policies on the devices. QoS mechanisms are good examples of methods affected by the policy provisioning, which again may affect the traffic flow in the network.

Policy-based routing (7)

To get control over the traffic flow in the network, the Policy-based framework uses predefined policies that are integrated with routing functions. This is called Policy-based routing. This may involve access control, resource allocation etc. For MANETs where the links are often of poor quality, it would be desirable to prioritise important traffic in front of other types of traffic, and eventually route this important traffic around bad links. This

Policy-based routing approach has been tested for a long time in wired networks, but there remains some work before the desirable effect could be obtained in MANETs.

Policy monitoring (8)

To distribute policies, the most important task is to configure devices with their policy specifications. However, we need to now if the devices in the network meet these policy specifications, and to ensure this a monitor mechanism will be needed. This could be achieved by using active packets like probes or passive packets like measurement-based estimations.

Adaptive Logic (9)

Due to the unstable behaviour in a MANET, there need to be some kind of coordination between the dynamic and the state-dependent policies. This must be done so that the current state of the network makes the control structure able to adapt. When a specified threshold is exceeded, a new type of policy needs to be used.

6.5.4 Discussion

The intension of the Policy-based framework management solution is to provide a robust and effective network in a lightweight manner. The greatest advantage of this solution is the ability to provide QoS. The network administrator may for instance assign specific network access for different devices, and divide resources among nodes after how important their traffic is etc. Policy-routing provides the network with the ability to route around paths with low bandwidth or high latency. This improves the possibility to use for instance VoIP that is very vulnerable to latency.

Management traffic in MANETs needs to be as low as possible. By the use of COPS-PR, distribution of policies is handled in an event based manner, and Policy-based management have thereby reduced the polling to a minimum and thus saved bandwidth.

The Policy-based management framework addresses the heterogeneity challenge in a satisfactory way. High-level policies are used for general network policies, while low-level policies are used to specify policies on different nodes or devices.

Today there remains some work to do regarding for instance Policy-based routing etc, before the mentioned method is ready to be implemented and used. There are however done some tests with the Policy-based framework with main focus on the clustering functions.

6.6 Overall discussion

The management methods proposed in this chapter have different ways to approach the task of managing a MANET. Some are more advanced than others, and some methods contain similar characteristics, like SNMP and ANMP. These methods use both the MIB II for data collection, but ANMP has in addition extended this MIB with an extra anmpMIB. This is done in order to collect information that is relevant for MANETs. Such information might be battery information etc. The Guerrilla management solution is just a management architecture, and it contains therefore no methods for data collection. It is designed to fit a variety of management protocols, which makes it usable in most management systems. The superior function of Guerrilla management compared to the other methods is its possibility to deliver management responsibility in a dynamic way between suited nodes. This makes it able to use the resources in the participating nodes in an economic way.

ANMP, Guerrilla and Policy-based framework contain all clustering algorithms to collect nodes in groups. This will make it much easier to solve routing problems, and it will give the manager less nodes to control. Clustering of nodes gives the network a better opportunity to deliver information in an event-based manner. This will reduce the overhead in the network, because the periodic polling is set to a minimum. SNMP needs to traverse all the nodes in the network in order to collect information, which is unhealthy as a network grows large.

SNMP deliver basic security in version 1, which is improved in version 3. This security is however the one least suited for MANETs. ANMP uses the security of SNMP v3, but has in addition added some new function for MANETs. However, the most suited solution for security issues would be the Policy-based approach. This method makes it easy to implement new security functions as they develop.

The ability to control the network behaviour in order to deliver some sort of QoS is important in MANETs, and Policy-based framework management is a good approach for such tasks.

7 Testing the SNMP protocol

7.1 Overview

In the previous chapter we discussed and evaluated different management solutions for use in a MANET. In this chapter we will test the chosen management solution (SNMP), in order to get real answers to the assumptions we made on this protocol. This is according to activity 4 in the task description (chapter 1.2). In addition we also tested software that was able to show the network topology. It was important for us to do the test in real environments, and not simulate the test. Because of the unpredictable nature of MANETs, this would give us much more realistic results. Also, both the influence from the physical layer and the link layer would be hard to measure in a simulated test.

Implementation and testing was a much larger task then anticipated. The reason for this is that much of the technology we used still is in an early phase. Doing real-tests inside of our technology-area is little explored and the documentation is insufficient. However, we put a lot of effort into this task and attacked the arising problems in a systematic way.

First in chapter 7.2 we will state the reasons for choosing the SNMP-solution for testing. In chapter 7.3 we will give an introduction to the testbed, which include describing the equipment we have used and the test-architecture.

We have divided the test into three parts, in chapter 7.4 we test a topology map to measure updating times, in chapter 7.5 we test the SNMP protocol as the data-load increases, and in chapter 7.6 we do a bottleneck test to see how fast it is discovered. Each of the tests is again divided into three parts. First we describe how the test was performed, next we present the results of the test, and finally we discuss the presented results.

7.2 Choosing the protocol

There where two main reasons for choosing the method we wanted to test in this Masters thesis, time and the status of the management protocol. As for guerrilla it is in the time being in the early stages of development and not ready for testing. ANMP is also in development but have been run on simulators, we did an approach to check out this method but showed to be unfit for real time testing at this moment. SNMP and Policy-based have both been around for a long time in wired networks, as for the Policy-based framework for wireless Ad Hoc is a very advanced approach and need a lot of implementing we decided to turn it down caused by the time available. This made us left with SNMP, this seemed like a good solution since it has been around for a long time and there are a lot of applications available.

7.3 Testbed description

7.3.1 Introduction

This chapter describes the equipment we use, both hardware and software, and the testbed topology.

7.3.2 Hardware equipment

To form a MANET we used five portable computers, which were all Linux-certified. Each of the computers was equipped with a wireless card supporting IEEE802.11b. The network was operating on a bandwidth of 2Mbps. We also used a stationary computer in some of the test, which had a wired connection to one of the portable computers.

7.3.3 Software equipment

The portable computers were running Linux RedHat7.3 as the operating system, while the stationary computer was running Windows XP. OLSR routing protocol was installed on all the portable computers, which required Linux as operating system.

To perform the topology-map test we only used the portable computers, giving one of them the task of management. On the management node we installed Cheops. This is Linux-based software showing all the computers in a network and how they are connected to each other. It is originally intended used in a wired network, but it worked well also for our type of network.

For the rest of the test we also used the stationary computer, which worked as the management node. For lack of good Linux-based SNMP-monitoring tools, we chose to

run a Windows-based SNMP tool called OiDView on the management node. This program can collect SNMP MIB-objects from the other nodes in the network, and view the collected information in graphs. The program assumes that the other nodes in the network are running an UCD-SNMP agent. The UCD-SNMP agent is collecting information from the node in which it resides, and responds to the requests done by the management node.

It was also necessary to use program which generated controlled traffic load. For this purpose we used RUDE&CRUDE. RUDE is the program running in the node that generates the traffic, while CRUDE is the program running in the node receiving the traffic. The traffic generated is UDP-packets. It is possible to set both packet-size and the number of packets generated each second.

We also use PING to measure the delay to each network node. This is a useful tool to see how this QoS-parameter behaves according to number of hops from the manager and according to the traffic load in the network increases.

7.3.4 Testbed architecture

The size of the testbed was limited to the number of nodes we had available, in this case five portable computers. We organized the nodes to form different topologies dependent on the test we wanted to carry out.

The topology we made use of the most, was the I-topology (figure 27). By organizing the computers this way, we utilized the networks full length. From node 1 to node 5 we had four hops or three intervening relay nodes. This topology was important, in order to test out the impact different number of relay-nodes had to the SNMP-traffic and network performance. In the tests were we used OiDView on the stationary computer, this had wired connection to node 1.

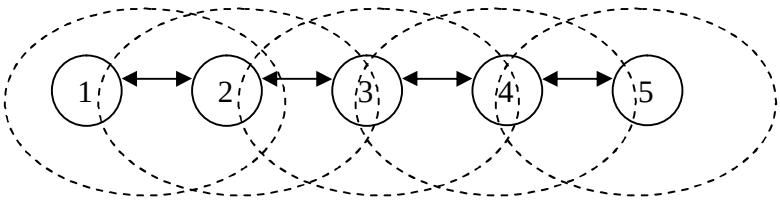

Figure 27 I-topology

For the topology-map test we also made use of two other topologies, Y-topology and O-topology.

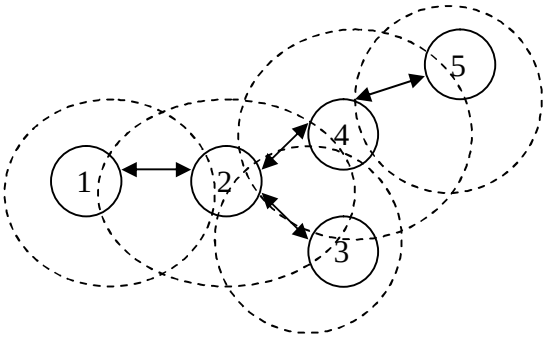

Figure 28 Y-topology

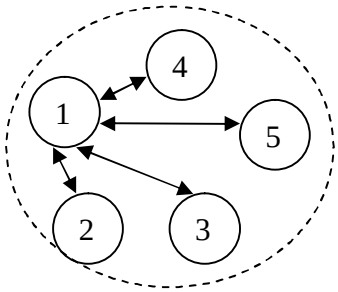

Figure 29 O-topology

7.3.5 Testbed environments

The test was performed in the corridors of HiA Grimstad, mostly at daytime. The traffic of persons going through the corridors and doors being opened and closed was therefore relative large and most variable. Such factors mentioned can have a great impact on the data transmitted, resulting in degraded and variable transmission conditions. The distance between the computers we placed throughout the corridors varied between 20 - 40 metres, dependent on walls and doors between them and the accessibility to wall outlets.

7.4 Topology test

7.4.1 Introduction

In this test we wanted to test topology map software running in a MANET environment. Monitoring the network topology is an important task in managing a MANET (further described in chapter 5.3). It is important that changes in the network topology are discovered within reasonable time, and that the task of doing this doesn't cause to much overhead in the network.

7.4.2 Topology software

To view the nodes in the network and the connections between them, we used a program called Cheops. This program is developed for a fixed wired network, and the automatic update of the topology map is limited to minimum one minute. When Cheops is updating the topology map, it first pings the network to search for hosts that are alive. The ping message is broadcasted onto the subnetwork specified in the software. To map the network and graph the connections between the nodes, it uses traceroute. To explain how traceroute works, we will introduce a small example scenario:

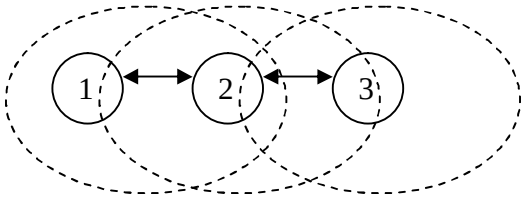

Figure 30 Three nodes in a MANET

Suppose we have the network above, where node 1 sends a traceroute with destination address equal to node 3. First traceroute sends an IP-packet with destination address equal to node 3, and Time-To-Leave- (TTL) field equal to 1. Node 2 receives this packet, decreases the TTL-field by one, and answers back to node 1 with its own IP-address. Because Node 1 didn't get answer from the destination node, it then sends a new IP-packet with the TTL-field equal to 2. The IP-packet is now received and answered by node 3, and traceroute is completed. Based on this we can see that the number of messages (caused by traceroute) sent from the node running Cheops, is equal to the number of hops a node is away from this node.

7.4.3 Test conditions

Because of the unsuitable interval for automatic updates of the topology map, the updating routine therefore had to be done manually. For all tests, we started manually updating the topology map immediately after taking down or up an interface. If the topology map did not show the expected topology after ten seconds, we did a new manual update of the topology map. The updating interval of ten seconds was chosen on background of the time the topology-map used to draw the whole network.

Node 1 is the management node in all of the topologies (see testbed architecture, 7.2.4).

In addition to the traffic generated by the Cheops-software, the routing protocol also generated traffic. OLSR, which was the routing protocol we used, sends two types of messages: Hello-messages and Topology Control- (TC) messages. Hello-messages perform the task of neighbour sensing and are set to a default interval of 0.5 seconds. TC-messages perform the task of topology declaration (advertisement of link states) and are set to a default interval of 2.0 seconds (for more information about OLSR, see chapter 2.3.3).

7.4.3.1 Scenario 1

In this scenario we used the Y-topology.

When we took node 2 down, the management node did not have contact with the other nodes in the network. However, the management node did have contact with all the other nodes when node 2 went up again.

7.4.3.1.1 Results

Table 2 Scenario 1 results

Attempt	Node 2 Up	Node 2 Down
1	19.0	6.2
2	17.4	6.3
3	16.9	6.2
4	17.7	6.2
5	17.6	6.2
6	17.6	6.1
7	17.8	6.1
8	17.5	6.2
9	17.7	6.1
10	16.8	6.2
Average:	17.6	6.2

7.4.3.2 Scenario 2

In this scenario we used the I-topology as a basis, but the number of nodes in the I-topology varied from one to five nodes.

When we took node 2 up, it only had contact with node 1.
When we took node 2 down, node 1 didn't have contact with any other node
When we took node 3 up, it only had contact with node 1 and 2.
When we took node 3 down, node 1 only had contact with node 2
Etc.

7.4.3.2.1 Results

Table 3 Scenario 2 results

Attempt	Node 5 Down	Node 4 Down	Node 3 Down	Node 2 Down
1	9.4	7.4	7.2	6.2
2	8.1	7.5	6.8	6.3
3	8.7	7.2	6.8	6.3
4	8.1	7.2	6.7	6.4
5	7.6	7.8	6.7	6.2
6	7.8	7.2	7.1	6.0
7	7.7	7.9	6.8	6.4
8	7.8	7.3	7.1	6.3
9	7.7	7.6	6.9	7.1
10	7.8	7.2	6.7	6.2
Average:	8.1	7.4	6.9	6.3

Attempt	Node 5 Up	Node 4 Up	Node 3 Up	Node 2 Up
1	19. 9	18.1	7.1	6.9
2	8.9	17.9	16.6	6.7
3	18.7	18.1	17.4	6.7
4	18.4	17.7	17.4	6.8
5	18.4	17.7	16.9	6.7
6	8.6	17.1	16.9	6.7
7	8.3	8.1	7.4	6.7
8	8.3	16.3	16.2	7.5
9	19.1	18.2	7.4	6.7
10	8.7	17.9	8.0	6.7
Average:	13.7	16.7	13.1	7.7

7.4.3.3 Scenario 3

In this scenario we used the O-topology.

The management node had only one hop to all the nodes in the network. We took all the nodes, except for the management node, up and down at the same time.

7.4.3.3.1 Results

Table 4 Scenario 3 results

Attempt	Node 2,3,4,5 Up	Node 2,3,4,5 Down
1	8.1	6.1
2	7.9	6.1
3	7.8	7.2
4	7.7	6.1
5	7.9	6.2
6	7.8	6.2
7	7.9	6.1
8	7.8	6.3
9	7.8	6.3
10	7.9	6.3
Average:	7.9	6.3

7.4.4 Discussion

There were two interesting perspectives with the time measures. First, did the number of nodes or the total number of hops away from the management node have impact on the time measures? And second, did the total number of new nodes and hops away from the management node have impact on the number of times we had to update the topology.

Before we discuss these questions, we will theoretical analyse what happens during a topology update. We updated the topology map at the same time as we took up or down a node. Because Cheops uses ping-broadcast to find all the nodes in the network, it is independent of the routing table. The nodes answer back with the originators IP-address. Traceroute, which is used to find the route from the manager to each of the nodes, is however dependent on an updated routing-table. It was therefore a race against time between the routing update and the Cheops-traceroute. If Cheops ran a traceroute to a new node, before the routing-table had added this new node, the topology map would not show the updated topology. When a node goes down, there will not be a problem showing the right topology, because the routing table does not have to be updated. Ping-broadcast will not reach the disconnected node and other nodes connected to this disconnected node.

Because all the time measures, updating of topology map and nodes going up and down are done manually, there will be an insecurity which might explain irregular results. Also considering that the data transmission is wireless, accomplices to potential variable results.

First we will see if there were any relations between the number of new nodes/hops and the probability that the topology map needed a second update after ten seconds.

Table 5 Relation between the number of new nodes/hops and the probability that the topology map needs a second update

Number of new nodes	Total number of new hops away from the management node	Number of second updates
1	1	0 out of 10
1	2	6 out of 10
1	3	9 out of 10
1	4	5 out of 10
4	8	10 out of 10
4	4	0 out of 10

From this table we can see that nodes lying one hop away from the manager didn't cause any second updates. This is also obvious since the management node is independent of the routing table to reach these nodes. New nodes more than one hop away from the management node caused however more second updates, which can be explained with the fact that the routing table was not updated fast enough.

When we further analyse the time measures we will not take into consideration that we needed a second update. The reason for this is that we want to see the impact number of nodes/hops had on the updating times. In other words, time measures that needed a second update were subtracted with ten seconds. The table below shows this relation.

Table 6 Relation between the numbers of new nodes/hops on the updating times

Number of nodes	Total number of hops away from the management node	Time/seconds
1	0	6.2
1	0	6.3
1	0	6.3
2	1	6.8
2	1	6.8
3	3	7.1
3	3	7.4
4	6	7.7
4	6	8.1
5	4	7.9
5	8	7.6
5	10	8.7

The minimum time Cheops uses to draw the network topology is about six seconds. This is a network consisting of only the management node. We can see a tendency that the updating time increases slightly with an increased total number of hops away from the management node. The maximum of 8.7 seconds were measured for the scenario with 5 nodes and a total number of hops away from the management node equal to 10. This seems reasonable since more packet traffic is needed to draw the topology map.

7.5 SNMP test

7.5.1 Introduction

Because of the limited bandwidth in a MANET, it is important that the SNMP traffic doesn't occupy too much of the total bandwidth in the network. We will therefore do some measurements of the overhead caused by this type of traffic. Another interesting perspective is to see what happens to the SNMP-traffic when the traffic load in the network increases.

7.5.2 Protocol stacks

Measuring the overhead caused by the SNMP-traffic, include looking at the protocol-stack to see what protocol headers that are necessary and how many bytes each of them require. For instance, when the management node wants to collect a MIB-object from another node, this request is wrapped up by the following headers:

802.11-header	IP-header	UDP-header	SNMP-header	Payload	Total packet size
34 byte	20 byte	8 byte	24 byte	GET request: variable	About 100 bytes or more

As we can see from the table above, an SNMP-GET message may occupy about 100 bytes of the network resources. A management node collecting this MIB-object once in a second, occupies therefore 0.8 kbit/s.

802.11-header	IP-header	UDP-header	SNMP-header	Payload	Total packet size
34 byte	20 byte	8 byte	24 byte	GET response: variable	About 100 bytes, but may be much bigger

The SNMP-GET request message is responded by a SNMP-GET respond message, which usually requires about the same number of bytes. However, the respond message may be much bigger dependent on the variables carried. RFC1157 [3] that deals with SNMP is not clear about the packet sizes.

The routing traffic has the following protocol stack:

802.11-header	IP-header	UDP-header	Payload	Total packet size
34 byte	20 byte	8 byte	Variable	Between 70-80 bytes

OLSR is sending HELLO-messages every half a second and TC-messages every two seconds. If each of these messages has an average of 80 bytes, the total bandwidth of these messages would be approximately 1.6 kbit/s. The routing traffic is broadcasted throughout the network and the messages are not responded. This is different compared to the SNMP-traffic, where nodes receiving SNMP-polling is responded back to the message originator.

To generate traffic load in the network, we used a program called RUDE&CRUDE. This program is generating UDP-traffic, with the possibility to determine the packet size and the number of packets to be sent each second. Because the 802.11 protocol has a maximum payload of 18496 bit, we didn't want to exceed this limit in order to avoid fragmentation of the packets generated by RUDE. The traffic load sent from RUDE had the following protocol stack:

802.11-header	IP-header	UDP-header	Payload	Total packet size
34 byte	20 byte	8 byte	1992 byte	2054 byte

We also made use of ping-traffic, to check out the delay in different nodes under different traffic load. Ping-traffic is wrapped up like this:

802.11-header	IP-header	ICMP-header	Payload	Total packet
34 byte	20 byte	4 byte	64 byte	122 byte

7.5.3 Test conditions

We used the I-topology during the whole test and node 1 was still the management node.

In scenario 1 we didn't run RUDE&CRUDE, meaning that the network traffic only consisted of routing-traffic and SNMP-traffic. The SNMP-traffic was generated from the management node, running OiDView. During the whole test, OiDView was polling a constant number of MIB-objects. The total number of MIB-objects were 24, 6 on each of the four nodes furthest out. These MIB-objects were on each node:

Interface objects: ifInOctets and ifOutOctets
SNMP objects: snmpInPkts and snmpOutPkts
UDP objects: udpInDatagrams and udpOutDatagrams

In addition to the MIB-traffic, OiDView was also sending ping traffic to each of the monitored nodes. This traffic had an interval of twice a second.

In the last two scenarios we also generated UDP-traffic with RUDE. The traffic was generated in node 5, i.e. the node furthest away from the management node, and collected with CRUDE in the management node. As measured above, each UDP-packet had the size of 2054 byte. We wanted to increase the network load from 0 % to 20 % to 40 % to

60 % to 80 % and finally 100 %, and then see what happened to the SNMP-traffic. We found that 20 packets/s = 320 kbit/s was about the maximum rate the network could forward, and this was therefore our 100 % network load. A higher packet rate than 320 kbit/s, didn't give a higher throughput in the other nodes.

In the last scenario we also sent some ping-traffic from the management node. We ran the ping-commando for about two minutes, and it pinged the destination node every second.

We must take into consideration that the transmission of data is wireless. Nodes that are within range of each other are sharing the same physical transmission medium. For instance, if node 3 is transmitting data to node 2 in the I-topology, also node 4 will receive this information. This means, if a packet is sent from node 5 to 1, the same packet will be received twice both in node 4 and 3. This is illuminated in the figure 31, where one arrow indicates one packet arrived. The number in the box above each node indicates the number of duplicates of each packet.

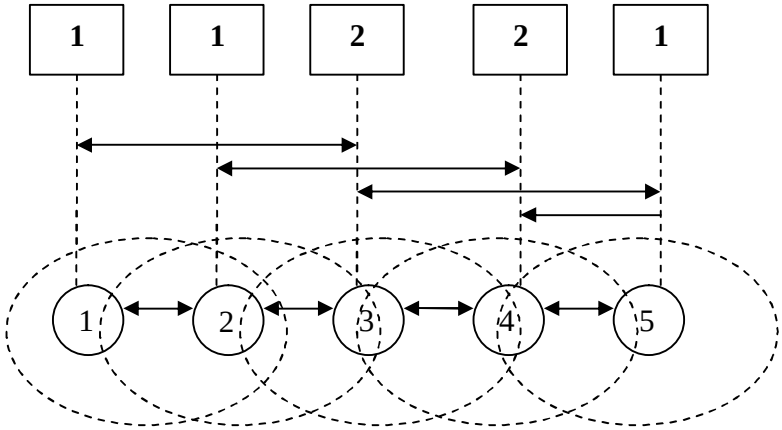

Figure 31 Overview of the nodes receiving duplicated traffic

The total traffic in and out of the wireless interface is countered by the MIB-objects ifInOctets and ifOutOctets on each of the nodes. These counters do not register such duplicated packets mentioned above, and is therefore not showing the real traffic going in the network. This must be taken into consideration when we perform the test and discuss the results.

It is important to notice that if OiDView fails in collecting a MIB-object, it just draw the last collected value for this object. In other words, the graph doesn't drop down to zero if it is unable to collect a MIB-object.

We will do a common discussion of all the three scenarios after describing each scenario and presenting the results.

7.5.3.1 Scenario 1

In this scenario we wanted to find out how much of the total bandwidth SNMP-traffic and routing-traffic were occupying. As mentioned above, we collected totally 24 MIB-objects. By monitoring ifInOctets (total traffic into the interface) and ifOutOctets (total traffic out of the interface) on each of the nodes, we could see the total traffic amount into and out of the wireless interface. Each of the colours on the graphs belongs to the following nodes:
Red: Node 2, Green: Node 3, Yellow: Node 4, Blue: Node 5

7.5.3.1.1 Results

X-axis: seconds, Y-axis: bytes

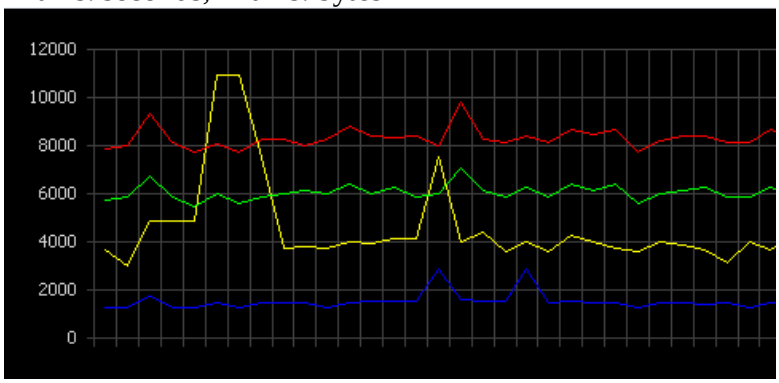

Figure 32 ifOutOctets on each node in scenario 1

X-axis: seconds, Y-axis: bytes

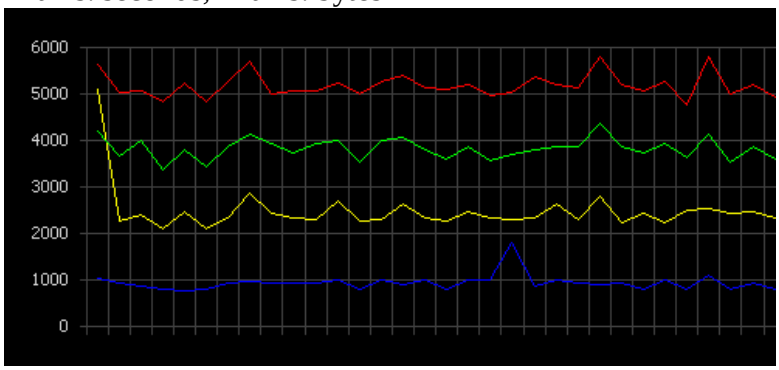

Figure 33 ifInOctets on each node in scenario 1

7.5.3.2 Scenario 2

After measuring SNMP-traffic and routing-traffic, we wanted to load the network with some RUDE-generated UDP-traffic. For every new measurement, we increased the traffic load with 4 packets/s = 64 kbit/s, and monitored both the traffic load and the SNMP-traffic in and out of each interface to see if they were showing the expected values. We will just present a selection of the measurements. The colours on the ifInOctets- and ifOutOctets-graphs belong to the same nodes as in scenario 1. The SNMP-packet graph has the following colour/node match:

Red: Node 2 In, Green: Node 2 Out, Blue: Node 3 In, Yellow: Node 3 Out, Purple: Node 4 In, Light blue: Node 4 Out, Grey: Node 5 In, Brown: Node 5 Out

7.5.3.2.1 Results

20 % extra load, 64 kbit/s

X-axis: seconds, Y-axis: bytes

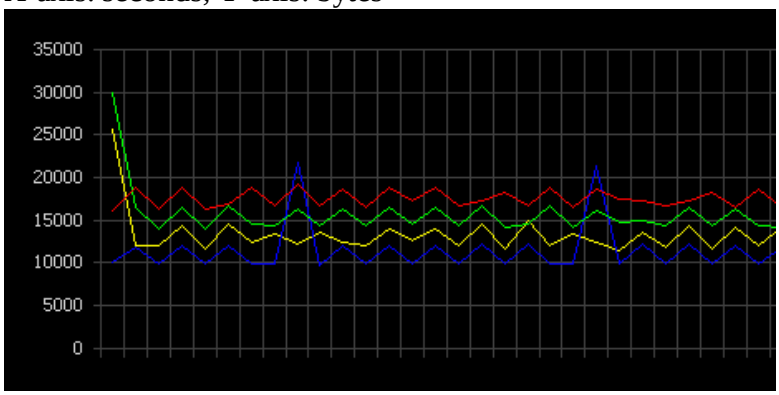

Figure 34 ifOutOctets on each node in scenario 2 with 20 % network load

X-axis: seconds, Y-axis: bytes

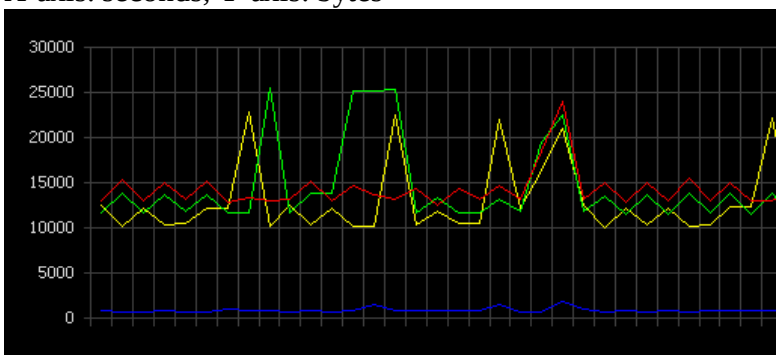

Figure 35 ifInOctets on each node in scenario 2 with 20 % network load

X-axis: seconds, Y-axis: SNMP-packets

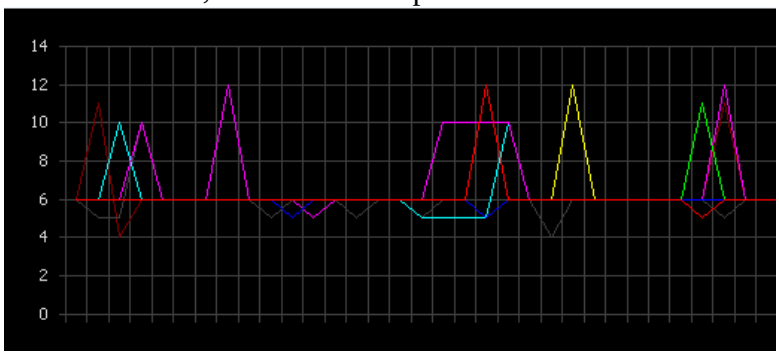

Figure 36 SNMP-packets on each node in scenario 2 with 20 % network load

100 % extra load, 320 kbit/s

X-axis: seconds, Y-axis: bytes

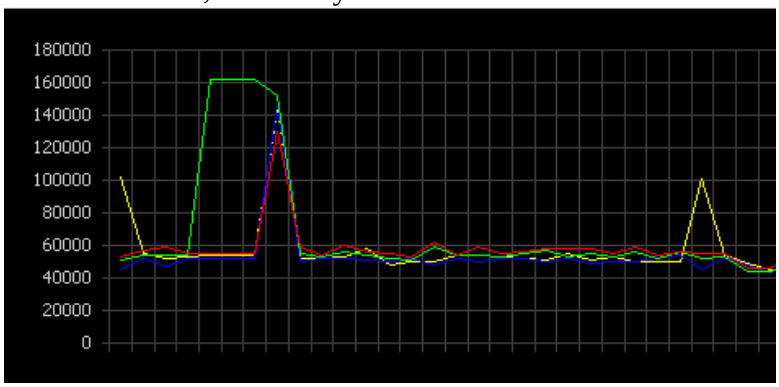

Figure 37 ifOutOctets on each node in scenario 2 with 100 % network load

X-axis: seconds, Y-axis: bytes

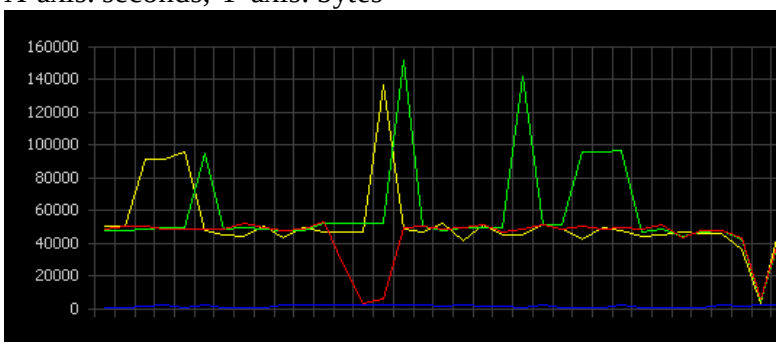

Figure 38 ifInOctets on each node in scenario 2 with 100 % network load

X-axis: seconds, Y-axis: SNMP-packets

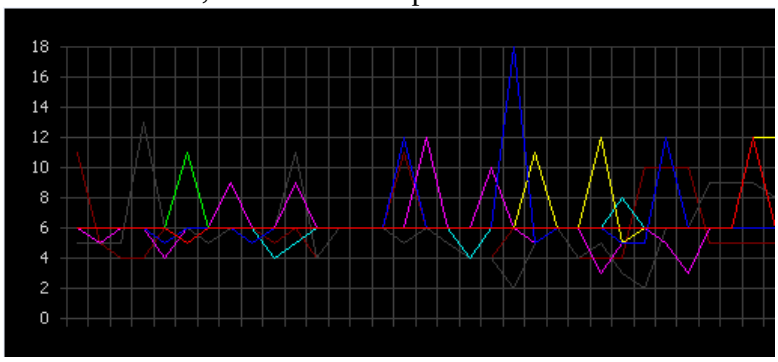

Figure 39 SNMP-packets on each node in scenario 2 with 100 % network load

7.5.3.3 Scenario 3

From the graphs in scenario 2 we could monitor the throughput in the network and also see how stable the SNMP-traffic was. In addition to the throughput-measuring, we also wanted to measure the delay to the different network nodes as the traffic load increased. The delay times are the time a ping-packet uses from the management node to the destination node, and back again to the management node. The traffic load in the network is the same as in scenario 2. Explanations to the tables: T = Transmitted packets, R = Received packets and L = Lost packets

7.5.3.3.1 Results

Table 7 Delay times to each node during different network loads

0 % load

Node	Min	Avg	Max	T	R	L
2	1.8	5.1	75.0	128	128	0 %
3	3.6	8.0	82.0	129	128	0 %
4	5.4	9.9	86.5	129	129	0 %
5	7.3	12.2	106.9	128	128	0%

20 % load

Node	Min	Avg	Max	T	R	L
2	1.8	7.3	112.3	119	119	0 %
3	3.6	13.3	109.6	128	124	3 %
4	5.5	25.4	267.8	117	114	2 %
5	7.3	18.0	208.7	142	137	3 %

40 % load

Node	Min	Avg	Max	T	R	L
2	1.8	5.5	88.7	138	134	2 %
3	3.6	13.7	126.8	120	115	4 %
4	5.4	18.8	167.9	130	130	0 %
5	7.3	21.8	160.8	136	134	1 %

60 % load

Node	Min	Avg	Max	T	R	L
2	1.8	7.9	125.3	125	122	2 %
3	3.6	19.7	213.7	125	124	0 %
4	5.4	20.5	174.1	125	121	3 %
5	7.3	30.2	343.3	126	124	1 %

80 % load

Node	Min	Avg	Max	T	R	L
2	1.8	10.3	142.1	128	128	0 %
3	3.6	15.6	162.7	131	126	3 %
4	5.4	31.9	668.9	129	125	3 %
5	7.2	20.5	156.2	124	117	5 %

100 % load

Node	Min	Avg	Max	T	R	L
2	1.8	5.6	76.0	127	125	1 %
3	3.6	35.1	376.4	127	125	1 %
4	5.5	87.2	441.9	128	124	3 %
5	7.3	137.6	659.1	129	127	1 %

7.5.4 Discussion

In a bandwidth limited network like the one in our testbed, it is important that the task of management does not occupy too much of the total available bandwidth. The bandwidth that the management traffic uses is of course dependent on several factors like:

- ? The number of MIB-objects to be collected from each node.
- ? The number of nodes in the network.
- ? The number of hops a node is away from the management node.
- ? How often the manager is polling the desired information.
- ? Management architecture
- ? If management information is collected by indirect polling (trap).

In our test we collected six different MIB-objects from four different nodes (node 2, 3, 4 and 5 in our I-topology), which totally make 24 MIB-objects. These MIB-objects were polled every second. In emergency situations like the one described in chapter 5.1 it is important that information about the network situation is up to date, which again include short polling intervals. SNMP uses typically a centralized architecture, which was the situation also for our MANET. Polling information from the network nodes in our I-topology, should therefore result in less SNMP-traffic the further away a node is from the management node. The reason for this is of course that SNMP traffic destined to nodes more than one hop away from the management node needs to be relayed. This theory agrees with the measurements done in scenario 1. The traffic out of the interface in node 2 has got the highest value (ca. 62.5 kbit/s), while the traffic out of the interface in node 5 has got the lowest value (ca. 11.7 kbit/s). When running with 100 % network load in scenario 2, the maximum throughput in node 2 was about 430 kbit/s. This means that in our MANET the SNMP-traffic together with routing-traffic took about 15 % of the total available bandwidth.

If we look on the ifInOctets-graph in scenario 1, we see that the traffic into node 5 is about 7.8 kbit/s. The traffic into node 5 (each second) will theoretically be the following:

- ? SNMP traffic (6 Get-request) = ca 4.7 kbit/s
- ? Routing traffic (2 Hello and 0,5 TC) = ca 1.5 kbit/s
- ? OiDView traffic (2 Ping) = 1.9 kbit/s

This would theoretical make a total traffic of 8.1 kbit/s, which matches the real results pretty good. If we compare the measured traffic into node 5 with the traffic out of this node, the difference is about 3 kbit/s. Because the duplicated traffic isn't registered in the SNMP agents, the number of packets in and out of the interface should be the same. The difference in the measured in and out traffic, can be explained by bigger responded data packets compared to the packets received. This passes mainly for the SNMP-traffic. Some of the difference can also be explained by some retransmission of packets.

In scenario 2, we increased the network load stepwise up to the maximum network throughput. By monitoring the total traffic in and out of each of the nodes interfaces, we could see if the graphs responded correct to the generated traffic load. Generating more then 320 kbit/s, did not give a higher throughput in the nodes forwarding this traffic. If

we look at how the SNMP-traffic responded to the increased traffic load, we can see a tendency to increased instability. Swings in the SNMP-graphs indicate that OiDView failed in collecting the information, in other words the packets got lost.

We also wanted to see how the delay varied between the different nodes under different traffic loads. Delay is an important QoS-parameter for many real-time applications, and must preferably be low with little jitter. Based on the results in scenario 3 we can see that the delay increased about linear for the number of hops away from the management node. The exception was for traffic load equal to 100 %, where the delay increased more exponential. Differences in delay between traffic loads from 0 % to 80 % were relative small.

The increased traffic load did not seem to have much influence to the packet loss of the ping-packets.

Our network was operating on a bandwidth of 2 Mb/s. The maximum and constant experienced throughput in our network was close to 430 kbit/s, which is about 0.43 Mb/s. This means that the total utilization of the network bandwidth was about 22 %. If we take into consideration all the duplicated traffic (described in chapter 7.5.3), the utilization of the network bandwidth is approximately twice as much.

7.6 Bottleneck test

7.6.1 Introduction

Because of the limited bandwidth in a MANET, bottlenecks are likely to occur. We therefore wanted to do a bottleneck test to see what happened to the SNMP-traffic and how fast the bottleneck could be discovered.

7.6.2 Test conditions

In this test we also used the I-topology. We started sending traffic from node 3 towards node 5. The traffic generated was equal to 96 kbit/s. This traffic rate did not cause any trouble for the network, and the throughput monitoring showed a stable graph in the implicated nodes. We wanted to create a bottleneck traffic, and started therefore sending a lot of traffic from node 5 towards node 3. This traffic load was equal to 800 kbit/s, a traffic load we knew would create a bottleneck in the implicated nodes (3, 4 and 5). We also took the time from the bottleneck-traffic was generated, to it was showed in the graphs on the management node. We used SNMP polling to discover bottlenecks. The colours on the graphs belong to the following nodes:

Red: Node 2 In, Green: Node 2 Out, Blue: Node 3 In, Yellow: Node 3 Out, Purple: Node 5 In, Light blue: Node 5 Out, Grey: Node 4 In, Brown: Node 4 Out

In = IfInOctets, Out = IfOutOctets

7.6.2.1 Results

Table 8 Time measures of discovered bottleneck-traffic

Attempt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Time/s	2.7	3.8	3.1	2.8	3.0	2.6	3.1	2.5	2.9	2.6	2.3	2.6	3.0	2.5	2.7

The average time from the bottleneck-traffic was generated to it was registered on the management node was 2.8 seconds

The graphs on the management node looked like this (two examples):

Example 1

X-axis: seconds, Y-axis: bytes

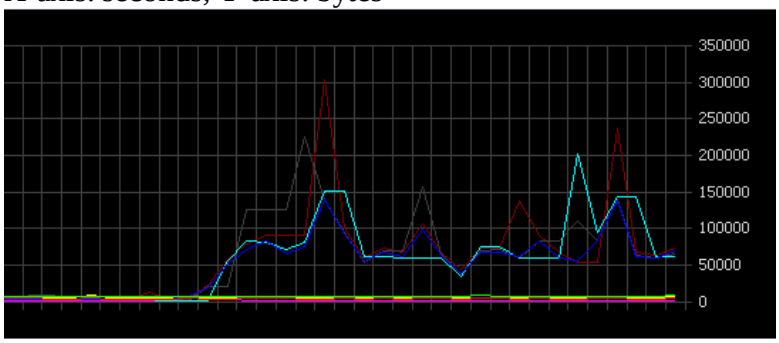

Figure 40 ifInOctets and ifOutOctets on each node in bottleneck test, example 1

Example 2

X-axis: seconds, Y-axis: bytes

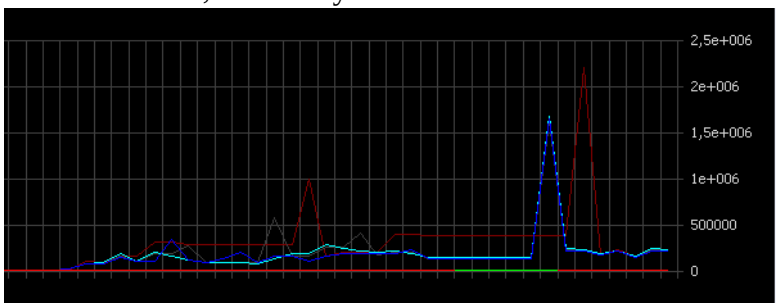

Figure 41 ifInOctets and ifOutOctets on each node in bottleneck test, example 2

7.6.3 Discussion

We measured the time elapsed from the bottleneck-traffic was generated to this traffic was registered on the management node (e.g. the graph started to increase). The average time was 2.8 seconds. After this time of period, the traffic increased for a few seconds up to its maximum throughput (about 430 kbit/s), and then the network got very unstable. The high traffic-peaks on the graphs can be explained by a big loss of SNMP-traffic. Before the highest peak on the example 2 graph, we can see that the brown line is constant for some seconds. This means that OidView is unable to collect IfOutOctets from node 4 for this time of period. Finally, when it is able to receive the variable, the IfOutOctets-counter has continued counting incoming octets, and has therefore got a very high value compared to the last time it was collected.

8 Discussion

8.1 Overview

Throughout this report we have made individual discussions on the theoretical parts and the practical parts. This chapter will try to discuss the most interesting theory and combine this with the results obtained from the tests. This is according to activity 5 in the task description (chapter 1.2)

In chapter 8.2 we will discuss the needs and challenges in managing MANETs, based on the theory from chapter 5. The different management challenges will be the foundation when we discuss the different management methods in chapter 8.3. The next chapter will do an overall discussion of the test results, and compare these results with the theoretical work made in chapter 5 and 6. Finally, in chapter 8.5 we will discuss interesting work to be done in the future.

8.2 Needs and challenges

Before considering how to do management of MANETs, we first have to analyse the needs for management of such networks. There are several factors that influence the significance of the management task. The number of nodes in the network, different requirements that the applications specify from the network, and the surroundings where the network is intended used are all such factors. However, because the main focus so far on MANETs has been limited to rescue- and military operations, the needs for management seems obvious. Common for such network scenarios are that the information flowing in the network is highly confidential and that the time factor is critical.

Management of MANETs has the same basic goals as management of wired networks. The main tasks are to monitor and control the network resources in order to make the network run effectively. However, the needs for management of MANETs differ somewhat from the one in wired networks.

Because MANETs are intended to be used in scenarios where a lot of confidential information is exchanged and often in hostile environments, the task of security management is very important. Network users need to be authenticated and the information sent needs to be encrypted in order to deny strangers in reading this information. Intruders trying to sabotage the network need to be discovered. The fact that the transmission of data is wireless, make the security issues even harder. Another need for management in MANETs is to monitor all the network nodes and their connections to each other. Together with a map of the particular area, the manager can monitor the networks coverage area. If the manager for instance discovers that a part of an accident area is without network connection, a new relay node may be placed out in order to increase the coverage area. Another example is a military operation where the number of nodes participating is known. By monitoring all network nodes, the manager can see how many nodes that are without connection to the rest of the network. If the number of nodes increases above the original number, this would imply that an intruder has been connected to the network. Together with the topology map, it would be of interest to monitor different parameters on each of the nodes. Especially QoS parameters like throughput, packet loss and delay, but also the battery level on each node would be of interest.

The characteristics of MANETs make them more challenging to manage than wired networks, and especially the task of providing QoS. Nodes in MANETs are free to move in an arbitrary manner, resulting in a highly changeable network topology. This might lead to network partitioning, and the sub-networks needs to be managed separately. Due to all the changes in network topology, a lot of traffic overhead is needed to keep an updated topology. Minimizing this overhead is a big challenge. Also, the fact that MANETs have limited bandwidth and variable link capacity makes the challenge of minimizing overhead extra important. The low and variable bandwidth also makes the QoS provisioning to a harder task. Guaranteed treatment of network traffic seems hard to

achieve, due to the dynamic topology and limited bandwidth. Because the network nodes are portable they need to run on batteries. This makes the network even more dynamic and therefore more challenging to manage.

There are mainly three management architectures, respectively centralized, distributed and hierarchical. Because of the characteristics of a MANET, the centralized management architecture seems like a poor solution, unless for a network with few nodes where the centralized architecture might be the most suitable. Nodes move frequently and the links are unreliable, network partitioning and nodes going down for a period of time are likely to occur. With one single management node, the management system could therefore be very unreliable and insufficient for parts of the network or even the whole network if the management node fails. The distributed architecture, where a group of autonomous managers collaborate in a peer-to-peer manner, makes the management task less vulnerable to faults and network partitioning. The scalability in such a network is however not optimal, even though better than the centralized architecture. A hierarchical architecture combines the two other architectures, and has therefore the advantages from both these architectures. This network is suited for large scalable networks.

8.3 Management solutions

The management solutions evaluated in chapter 6 solve the different challenges in MANETs with different degree of success. We will look at the most important challenges, and discuss the different management solutions up against these challenges.

8.3.1 Dynamic topology

SNMP, our chosen management solution for implementation and testing, is originally intended used in wired networks, and have no good approaches for solving a rapidly changing environment. It is based on point-to-point connection between the manager and the managed nodes. ANMP, Guerrilla and Policy-based have taken the rapidly changing topology into consideration by introducing clustering algorithms. The network nodes are grouped together in clusters, and each cluster is controlled by a sub-manager. The manager node then just needs to communicate with each of the sub-managers, which results in better control for the manager when topology changes occur.

8.3.2 Low bandwidth and variable link quality

Because of the low bandwidth and variable link quality it is important to minimize the traffic overhead. All solutions presented in this document have event based function to ease this problem. SNMP have trap messages to inform the top manager of critical event happenings. However, they are limited to a fixed number, but can give the manager valuable information of the network behaviour and reduce the overhead due to less polling. Compared to SNMP, ANMP is extended with an anmpMIB that collect more Ad

Hoc relevant data, like battery capacity etc. Also, network clustering reduces the management traffic in the network considerably, since the manager does not have to be in direct contact with every node. Guerrilla is just an architecture proposal, and can use most management protocols for the collection of data. This makes it able to optimise the collection process that is best suited for the situation the network is used. It also supports clustering to ease the data collection and minimize the overhead between the manager and the managed nodes. The Policy-based approach is mainly for controlling the network by using policies, and do not collect network information. The different network nodes contact the manager in order to get their behaviour policies when they are initialised in the network. After that they will just send requests to the manager when special events occur. Policy-based management is strongly designed to make the network nodes autonomous by the use of policies. The nodes must know what to do when for instance critical events occur, which will reduce the management traffic considerably.

8.3.3 Limited resources and heterogeneity

The limited resources in the devices attending a MANET and their heterogeneity, is a difficult problem to solve. Standard SNMP-solutions deliver no support for battery monitoring, but this may be possible to implement in the experimental branch in the MIB. Without a battery-MIB, the different nodes may just disappear without further notice caused by battery draining. SNMP is a very simple and lightweight protocol and there are solutions for most operating systems, which makes it very flexible in use. ANMP is much like SNMP, but it has added among others some MIB-objects for monitoring status of the battery power. Today's ANMP solutions is just for Linux operating systems, and have not been tested in a real environments. It has inherited the simplicity of SNMP and should with further development be able to function on most nodes. The Guerrilla architecture stands out from the other solutions. The responsibility of being a manager node will constantly be delegated to the node best suited for this task, which may be the one with highest CPU, most battery power left etc. This method exploits the different nodes available resources to the maximum.

8.4 Test

The use of Cheops in our test delivered us satisfactory management information, but the software had some weaknesses for use in MANETs. The use of broadcast ping to discover the network nodes and then running a traceroute to map these nodes is not an optimal way of monitoring the network topology. Because of the rapidly changing topology in MANETs, updates needs to be done often in order to show the correct topology. Doing a broadcast ping and a traceroute to each of the nodes every time the topology map is updated would add a lot of traffic to the network. The SNMP system group in MIB2 consists of a topology objects that contains information about the surrounding nodes. By polling the nodes for their topology tables, this would reduce the traffic load. The most economic way of updating the topology map would be to send trap-messages every time a node experiences changes in its routing table. We would then

avoid the broadcast pinging and the traverse of the network with traceroute to obtain the necessary network information. Another backside with the traceroute function is that it only finds one route from the manager to the different nodes. For instance in our Y-topology, the connection between node 4 and 5 was not able to be discovered.

In the evaluation of SNMP we forecasted that SNMP would produce a large amount of overhead in the network caused by its polling and its architecture. The polled information needs to traverse all the intervening nodes between the manager and the managed node. Based on the measurements done in our test, we found that SNMP (together with routing) used 15 % of the total bandwidth. We used relative few nodes in our test, and as the number of network nodes increases, the total SNMP traffic would just increase and finally lead to an "over" managed network. This means that the task of management is taking so much of the total bandwidth that it destroys for its intended purpose, namely an effective network. To minimize the overload traffic in our test, we could for instance change the polling interval to 5 seconds or higher. When the traffic load approached the maximum throughput in our test network, the SNMP traffic got more unstable. The graphs of the network traffic showed high peaks, which was a result of SNMP packet loss in the network. Collecting much management information when the traffic load is close to the maximum will not be a good idea, and reducing the polling interval or reducing number of MIB-object might be useful.

We also did a test to check how fast bottlenecks were detected and to see how the management traffic then responded. We found that it took an average of 2.8 seconds from the bottleneck-traffic was created to it was registered on the management node. This is a reasonably fast report time. It took another two or three seconds before the traffic had exceeded the maximum throughput and a bottleneck was discovered. In our test we suddenly increased the traffic stream to an amount we new would be far too high and cause an immediate bottleneck situation. This is not likely in a real world scenario where the traffic is more likely to increase more slowly towards a critical traffic amount. SNMP have the possibility to send trap messages when for instance the throughput exceeds thresholds of 70 % and 80 % of the maximum throughput. When the nodes are sending such trap-alarms, the administrator would maybe have time to react and might solve the traffic problem before the nodes gets congested.

When we increased the traffic load from 0 % to 100 %, we did not experience much increase in the packet loss. Also the difference between the total traffic out and in of each node stayed relative constant. This means that the amount of retransmissions was not worth mentioning. However, when we were generating the bottleneck traffic we experienced high packet loss in the congested nodes, resulting in retransmissions and degraded SNMP- and routing performance.

8.5 Further work

“The purpose of this working group is to standardize IP routing protocol functionality suitable for wireless routing application within both static and dynamic topologies”. [1] This statement is collected from the MANET working group. In other words, the task of managing MANETs is not one of their working areas. However, in chapter 6 we discussed different approaches for managing such networks. Both for ANMP and Guerrilla there has not been any further studies or real implementations. ANMP has been tested in a simulated environment and in the case of Guerrilla, there has been tested a very simple prototype to show the migration of the nomadic manager. Policy-based management have been the working area for a group of people for some time now, and their future work includes complete implementation of their framework. Also the task of Policy-based routing is interesting for further investigation.

As far as we know, there have not been done similar real-tests of SNMP in a MANET. We were doing the testing in a relative small network consisting of five nodes. It would have been interesting to test the SNMP protocol on a larger network, and with more heterogeneous nodes. Some outdoor tests with moveable nodes would have given even more realistic results. Another interesting test is to see how much the use of trap can reduce the management overhead and still provide the necessary management information.

Testing the other management solutions in a real environment would have given a basis for comparing the solutions up against each other. Comparison of the management overhead, stability of the network, the clustering algorithms and the ability to provide QoS management are some of the interesting factors.

Because of the relative small effort made on the topic of developing management solutions for MANETs, there is still much work to be done in the future. Both when it comes to development and testing of existing management solutions, and when it comes to considering and development of new management solutions.

9 Conclusion

We have in this Masters thesis evaluated needs and solutions for management of MANETs, with main focus on QoS. The evaluation of the management solutions was first done theoretically, and then the best available solution was tested in a real environment. Even though we experienced some variable results during the test of the SNMP solution, we found a strong correlation between the theory and the test-results.

There is definitely a large need for management in MANETs. Typical scenarios where MANETs are intended used are disaster- and military-operations. A lot of sensitive and essential data exchanged between the network nodes, are common for such scenarios. The task of management is therefore important in order to keep a secure and effective network. Monitoring and controlling the network topology and different performance parameters like throughput and delay are necessary in order to perform QoS management.

The task of network management and in particular QoS management in MANETs is a very challenging task. Most management solutions available today, do not have sufficient support for this. SNMP and ANMP are protocols well suited for monitoring traffic, but they have only limited possibilities to dictate the behaviour of the network nodes. The extended framework for Policy-based management for MANETs deals with this issue in a more or less satisfactory manner. Distributing policies to the different network nodes make them more autonomous and capable of resolving problems within themselves. This will also result in more economic utilization of bandwidth, due to less overhead. However, this approach does not have the same possibility as SNMP and ANMP to monitor the network nodes.

In our test of SNMP we experienced relative high overhead, and with a high increase of network nodes, SNMP would not be a good alternative. However, for networks containing few nodes, like the one in our test, SNMP seems like a good alternative. For larger networks it would be desired to have a more event-based solution, where each node takes initiative to tell the mangers about desired network behaviour. Clustering of the network is also a requirement for large networks. This distributes the task of management among more nodes, and reduces the overhead and vulnerability in the network.

ANMP would be a good approach for traffic monitoring, and in coexistence with the extended framework for Policy-based management, the manager would also be able to control the network and provide the desired QoS management.

10 Abbreviations

AODV	Ad Hoc On demand Distance Vector
ANMP	Ad Hoc Network Management Protocol
ASN.1	Abstract Syntax Notation 1
BER	Basic Encoding Rules
CCITT	the united nations Consultative Committee for International Telephony and Telegraphy
CEDAR	Core Extraction Distributed Ad hoc Routing
COPS	Common Open Policy Protocol
COPS-PR	COPS for Provisioning
DEC	DECision message
DCF	Distributed Control Function
DCM	Data Collection Module
DiffServ	Differentiated Services
DSSS	Direct Sequence Spread Spectrum
DynaSeR	Dynamic Service Redundancy
ECN	Explicit Congestion Notification
EDCF	Enhanced Distributed Coordination Function
EENM	Execution Environment for Nomadic Manager
EGP	Exterior Gateway Protocol
FCAPS	Fault- Configuration- Accounting- Performance- Security-management
FDMA	Frequency Division Multiple Access
FIFO	First In First Out
FHSS	Frequency Hopping Spread Spectrum
FQ	Fair Queuing
FQMM	Flexible Qos Model for Manets
GFSK	Gaussian Frequency Shift Keying
GMIB	Guerrilla Management Information Base
HCF	Hybrid Coordination Function
HiA	Høgskolen i Agder
IAB	Internet Architecture Board
ICMP	Internet Control Message Protocol
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
iMAQ	integrated Mobile Ad Hoc Qos Framework
IntServ	Integrated Services
IP	Internet Protocol
ISO	International Standards Organization
LDAP	Lightweight Directory Access Protocol
MAC	Media Access Control
MACA/PR	Multihop Access Collision Avoidance with Piggyback Reservation
MANET	Mobile Ad Hoc NETworks
MIB	Management Information Base

NMM	Nomadic Management Module
NMS	Network Management System
OID	Object IDentifier
OLSR	Optimized Link State Routing
OS	Operative System
OSI	Open Systems Interconnection
PAN	Personal Area Network
PC	Personal Computer
PDA	Personal Digital Assistant
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PDU	Packet Data Unit
PPM	Probe Processing Module
PQ	Priority Queuing
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RED	Random Early Detection
REQ	REQuest
RFC	Request For Comment
RIO	RED with In and Out
RMON	Remote network MONitoring
RSVP	Resource Reservation Protocol
SMT	Structure of Management Information
SNMP	Simple Network Management Protocol
SWAN	Stateless Wireless Ad Hoc Networks
TC	Topology Control
TCP	Transport Control Protocol
TDMA	Time Division Multiple Access
TTL	Time To Leave
UDP	User Datagram Protocol
USM	User-based Security Model
VACM	View-based Access Control Model
VoIP	Voice over IP
WFQ	Weighted Fair Queuing
WLAN	Wireless Local Area Network
ZRP	Zone Routing Protocol

11 References

[1] IETF MANET Working Group
http://www.ietf.org/html.charters/manet-charter.html

[2] Wenli Chen, Nitin Jain and Suresh Singh
ANMP: Ad Hoc Network Management Protocol August, 1999
http://www.cs.pdx.edu/~singh/papers.html

[3] IETF Network Working Group
A Simple Network Management Protocol (SNMP), RFC 1157, May 1990
http://www.ietf.org/rfc/rfc1157.txt

[4] Chien-Chung Shen, Chaiporn Jaikaeo, Chavalit Srisathapornphat and Zhuochuan Huang
The Guerrilla Management Architecture for Ad Hoc Networks
http://alfalfa.cis.udel.edu:8080/refs/papers/shen02Guerrilla.pdf

[5] Kaustubh S. Phanse, Luiz A. DaSilva and Scott F. Midkiff
A Taxonomy and Experimental Evaluation of Policy Architectures for Bandwidth-constrained Networks, May 2002

[6] Kaustubh S. Phanse and Luiz A. DaSilva
Addressing the Requirements of QoS Management for Wireless Ad Hoc Networks, 2002

[7] Kaustubh S. Phanse, Luiz A. DaSilva and Scott F. Midkiff
Extending Policy-based Management to Ad Hoc Networks, 2003
http://www.ee.vt.edu/~kphanse/Adhoc_policy_mgmt.pdf

[8] Kaustubh S. Phanse
Policy-based Quality of Service Management in Wireless Ad Hoc Networks, October 2002
http://www.ee.vt.edu/~kphanse/kphanse_prelim_document.pdf

[9] IETF Network Working Group
Mobile Ad Hoc Networking (MANET), RFC 2501, January 1999
http://www.ietf.org/rfc/rfc2501.txt

[10] Mohammad Ilyas
The handbook of Ad Hoc wireless networks, 2003

[11] C. K Toh
Ad Hoc Mobile Wireless Networks protocols and Systems, 2002

[12] Kanoksri Sarinnapakorn
IEEE 802.11b "High Rate" Wireless Local Area Networks, March 2001
http://alpha.fdu.edu/~kanoksri/IEEE80211b.html

[13] palowireless
Bluetooth Tutorial – Specifications
http://www.palowireless.com/infotooth/tutorial.asp

[14] IETF Network Working Group
An Architecture for Differentiated Services, RFC 2475, December 1998
http://www.ietf.org/rfc/rfc2475.txt?number =2475

[15] IETF Network Working Group
Integrated Services in the Internet Architecture: an Overview, RFC 1633, June 1994
http://www.ietf.org/rfc/rfc1633.txt?number =1633

[16] AVLSI
Channel and QoS Adaptive Multimedia Wireless Ad Hoc Networks
http://vlsi.cornell.edu/MURI/pld.html

[17] Prasant Mohapatra, Jian Li and Chao Gui
QoS in Mobile Ad Hoc Networks, March 2003
http://www.cs.ucdavis.edu/~prasant/pubs/journal/manet-survey.pdf

[18] Hannan Xiao, Winston K.G. Seah, Anthony Lo and Kee Chaing Chua
A Flexible Quality of Service Model for Mobile Ad-Hoc Networks
http://www.cwc.nus.edu.sg/~cwcpub/zfiles/fqmm.pdf

[19] Seoung-Bum Lee, Gahng-Seop Ahn, Xiaowei Zhang, and Adrew T. Campbell
INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks, December 1999
http://comet.ctr.columbia.edu/~campbell/papers/jpdc.pdf

[20] Gahng-Seop Ahn, Adrew T. Campbell, Andreas Veres and Li-Hsiang Sun
Supporting Service Differentiation for Real-Time and Best-Effort Traffic in Stateless Wireless Ad Hoc Networks (SWAN), July-September 2002
http://comet.ctr.columbia.edu/swan/swan-tmc.pdf

[21] MONET Research Group
iMAQ: An Integrated Mobile Ad-hoc QoS Framework
http://cairo.cs.uiuc.edu/adhoc/

[22] Zeinalipour-Yazti Demetrios
A Glance at Quality of Services in Mobile Ad-Hoc Networks
http://www.cs.ucr.edu/~csyiazti/courses/cs260/html/manetqos.html

[23] Cisco Systems
Network management basics, Chapter 6
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/

[24] Allan Leinwand and Karen Fang Conroy
Network Management, A Practical Perspective, 1996

[25] FutureSoft
FCAPS White Paper
http://www.futsoft.com/pdf/fcapswp.pdf

[26] Uyless Black
Network Management Standards, The OSI, SNMP and CMOL Protocols, 1992

[27] Dr. Sidnie Feit
SNMP a guide to network management, 1995

Support literature:

Berner Vegge
Mobile Ad-Hoc Scenarios, Munin-Applica document, February 2003

Helge Gundersen and Frode Trydal
QoS for real-time IP traffic, Graduate Thesis for Agder University College, May 2001

MANET Mailing List

Cisco Systems
Quality of Service Networking
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm

Steven T.Joyce and Jon Q. Walker
Getting Started with QoS and Policy-based Network Management