WMM - Wireless Mesh Monitoring

Ricardo Pinto
ricardo.pinto@tagus.ist.utl.pt

Instituto Superior Técnico
Advisor: Professor Luís Rodrigues

Abstract. Wireless Mesh Networks (WMN) have emerged as a potential technology to quickly deploy a wireless infrastructure that is self-healing, self-configured, and self-organized. This report makes an introduction to WMNs, their protocols and some existing deployments. Then the problem of monitoring the operation of these networks, is addressed by making a brief survey on some relevant network monitoring approaches and by pointing directions of future work in this area.

1 Introduction

Within the short span of a decade, wireless networks have revolutionized the way we use our devices, bringing us cable-free mobility, always-on connectivity, and reduced infrastructure costs. In the past few years, we have witnessed a tremendous growth of wireless LANs (WLANs) mainly due to their ease of deployment and maintenance. However, all wireless access points (APs) need to be connected to the wired backbone network and, therefore, WLANs still require extensive infrastructure and careful planning in order to minimize their costs. Wireless Mesh Networks (WMNs) have emerged as a key technology to ease the deployment of wireless networks. Unlike traditional WiFi networks, in WMNs only a subset of APs are required to be connected to the wired network. As a result, only a subset of the nodes need wired infrastructure (these nodes serve as gateways) while the other mesh nodes have the ability to route messages to the gateways, thus providing also access to the Internet. A WMN is a dynamically self-organized and self-configured network in which the nodes automatically establish and maintain connectivity. These features allow a low-cost network, ease to deploy, and offering scalable coverage. Recently, a significant amount of research focus has been directed towards the study and deployment of WMNs.

As in any other network, an important activity that needs to be supported in WMNs is network monitoring, in order to allow operators to gather information about the network operation and quickly detect anomalies or performance degradation. Unfortunately, network monitoring requires the exchange of information in the network and is also a source of overhead. If performed incorrectly, network monitoring traffic may have a negative impact on network performance. Therefore, it is important to use the most adequate monitoring solutions, that minimize the consumption of network resources. As in every network monitoring

system, a trade-off must be achieved between the information freshness and the network resources consumed.

This report makes a survey on the operation of WMNs and of some relevant monitoring techniques that can be applied to these systems. It then sketches a proposal for the implementation of a monitoring system for WMNs and addresses the techniques that may be used to evaluate its performance, using both simulations and a experimental deployment.

The rest of this report is organized as follows. Section 2 describes the goals and expected results of this work, Section 3 provides a survey of the related work. Section 4 provides a sketch of the monitoring system we plan to implement and Section 5 discusses how it can be evaluated. A schedule of future work is given in Section 6 and finally Section 7 concludes the report.

2 Goals

This work addresses the problem of network management of WMNs, with emphasis on the monitoring of the network operation. More precisely:

> *Goals:* This works aims at designing, implementing, deploying, and evaluating a monitoring tool for WMNs.

In detail, this work aims to provide a generic monitoring and testing framework for routing protocols in WMNs with the following characteristics: protocol independent; modular and extensible; supporting accurate statistical measurements of network traffic; scalable and; CPU and bandwidth-efficient. We plan to implement a prototype of the tool and evaluate it in a real deployment, using a mesh of wireless routers. In the end, we expect to achieve the following results.

> *Expected results:* i) a fully functional implementation of the monitoring tool; ii) an evaluation of the performance of the monitoring mechanisms, in particular of their overhead, using simulations; iii) a practical evaluation of the tool, based on a real testbed.

3 Related Work

This section provides a survey of the related work. Subsection 3.1 provides a brief introduction to WMNs. Subsection 3.2 describes some of the routing protocols that are currently used in WMNs. Subsection 3.3 describes the main techniques to evaluate WMNs and Subsection 3.4 provides an overview of some existing testbeds. Finally, Subsection 3.5 addresses the existing monitoring solutions for WMNs.

3.1 Wireless Mesh Networks

Mesh networking has its roots in tactical military networks, comprised of nodes with multiple interconnections that stored and forwarded packets[1]. Attracted by the inherent survivability and robustness of mesh networks, the US Defense research agency DARPA funded several projects that support troop deployment on the battlefield. PRNET[2] project was started in 1973 and was a multi-hop Packet Radio NETwork system that reached a size of 50 nodes, allowing some to be mobile. More recently, the IEEE (its 802.11 Working Group) has tackled the standardization for wireless mesh networks. The 802.11s standard[3,4] specifies the physical (PHY) and medium access control (MAC) layers and a default mandatory routing protocol: Hybrid Wireless Mesh Protocol (HWMP), although it allows alternate protocols to be used. With the broad availability of WLAN hardware and small-scale, low-cost portable devices in the late 1990s, interest in these networks increased dramatically.

Characteristics A WMN is a multi-hop network, dynamically self-organized, self-configured, self-healing, resilient to device failures, and highly scalable; where all the nodes in the network assure the availability of one or more paths among different nodes in the network[5][6]. In detail, WMN should have the following properties:

Multi-hop wireless network A WMN extends the coverage of current wireless networks without sacrificing channel capacity, because intermediate routers forward each other's traffic and provide non-line-of-sight (NLOS) connectivity. Also, by correctly placing the nodes it can achieve an efficient frequency re-use.

Dynamically self-organized, self-configured and self-healing Each node that joins the network, automatically establishes connections to other nodes without previous configuration needed. Adding new nodes or even relocating them is as simple as plugging them on. The mesh is self-healing precisely because no human intervention is necessary for rerouting messages.

Resilience to device failures This can be achieved by rerouting the packets around the failed nodes. Since for a given pair of nodes, the probability of having two or more routes inter-connecting them is high, if one (intermediary) node fails, the network will adapt and route the packets through different paths.

Highly scalable The WMN should be expandable, allowing to add more routers in order to support more clients and cover a wider geographical region. The scalability depends on factors such as the size of the network, its architecture, topology, traffic pattern, node density, number of channels, and transmission power, among others. When the system grows it is desirable to have more gateways, given that the lack of the appropriate number of gateways may cause traffic bottlenecks and reduce network performance.

WMNs have the potential to help users to be always online, anywhere, anytime. Moreover, the gateway functionality enables the integration of WMNs with

several existing networks such as cellular, wireless sensor, WiFi, WiMAX and wired networks. Conventional nodes such as laptops, PDAs and phones equipped with wireless network cards can connect directly to the mesh network.

Network Architecture Typical WMNs are comprised of two network components[7], as described below and illustrated by Figure 1:

Mesh Routers Mesh routers can be divided into gateways and backbone routers. Gateways are connected to the wired network and support transparent bridging and address learning. Backbone routers provide mesh services and may be a typical Access Point (AP) to which clients connect, or a dedicated infrastructure device that only enhances network coverage and capability (no AP services).

Mesh Clients Mesh clients can be routing or non-routing capable. If the clients can forward network packets, then the network coverage can be further increased just by having more clients connected to the mesh routers and to each other.

The configuration of a WMN has to be carefully planned and there are three possible scenarios[1], depicted in Figure 2, according to the characteristics of the hardware:

- The first generation of WMNs uses only one radio channel to provide client access and backhaul service. This is the worst of all options since both clients and backhaul compete for bandwidth, and nodes have to listen, then send, and then listen again; this intermittent behavior affects network performance.
- The second generation adds one more radio, separating backhaul and client service networks. The non-overlapping of both networks frequency-wise, improves performance when compared to the first generation. Still, a single radio frequency is servicing the backhaul, traffic destined to external networks shares bandwidth on each hop leading to network performance degradation (not as severe as first generation).
- The third generation dynamically manages channels of all radios in order to avoid channel interference. In this 3-radio configuration, two radios provide the up and downlink of the backhaul and the other radio provides service to the clients.

Application Scenarios A good example of a mesh application is home networking. Nowadays, most wireless coverage is provided by 802.11 WLANs. In these networks, the localization of the access points can cause dead zones without service coverage. An approach based on a WMN can provide a full and flexible house coverage.

[1] http://www.dailywireless.org/2004/07/15/the-mesh-debate/

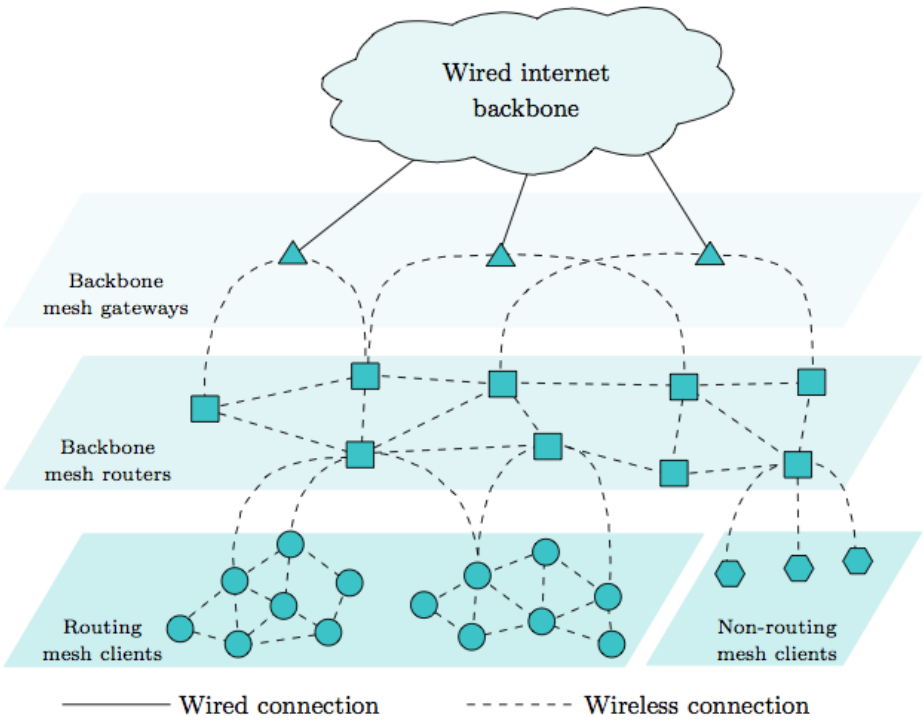

Figure 1. Layered Network Architecture.

The 802.11 WLANs are also widely used in enterprise offices, and the Ethernet cabling is a key reason for the high cost of the wired infrastructure in small and medium enterprises. APs can be replaced by mesh routers that provide client connectivity and cable-free backhaul.

WMNs can also be applied to transportation systems where remote in-vehicle video and driver communications can be supported; domotics, where WMN can help reduce the cost of wired networks to manage lifts, power, lights and AC; and security surveillance of areas such as parking lots, shopping malls and grocery stores. In addition to the above scenarios, WMNs can also be applied to emergency situations where the simple placement of wireless mesh routers can quickly establish connectivity[8].

3.2 Protocols

WMNs are unstructured networks, and protocols have to account for mobility, dynamic changes in topology, and the unreliability of the medium. WMN nodes communicate with each other and routes to non-neighboring nodes have to be

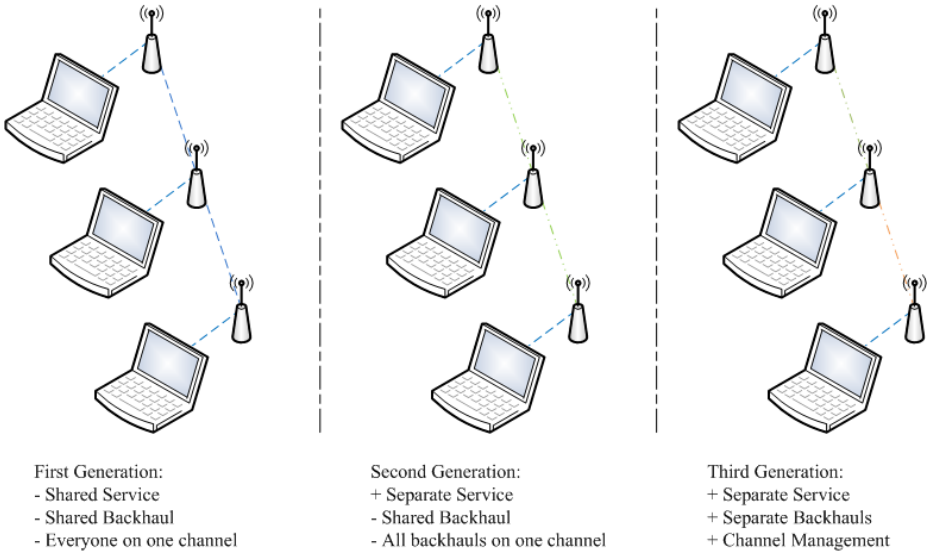

Figure 2. Mesh Configurations.

established. Routing protocols are responsible for discovering, establishing and maintaining such routes. Routing protocols for WMN are mostly based on protocols designed for mobile ad hoc networks. These can be classified in the following categories[9]:

Proactive protocols construct the routing table periodically. Each node maintains a table representing the entire network topology which is regularly updated in order to maintain the freshness of routing information. At any given time, any node knows how to reach another node of the network. This approach minimizes the route discovery delay at the cost of exchanging data periodically, that consumes network bandwidth.

Reactive protocols construct the routing table on-demand. Nodes are not aware of the network topology and find routes by flooding the network with route requests. This leads to higher latency due to the fact that the route has to be discovered, but minimizes control traffic overhead.
Usually, reactive protocols are better suited in networks with low node density and static traffic patterns. Since the traffic patterns are static, the first request encompasses the route discovery, while the subsequent use the previous discovery to route traffic. On the other hand, proactive protocols are more efficient in dense networks with bursty traffic, due to the continuous exchange of topology information, reducing route discovery delay.

Hybrid protocols are a mixed design of the two approaches mentioned above. These protocols typically use a proactive approach to keep routes to nodes in the vicinity of the source, but for nodes beyond that area, the protocol

behaves like a reactive one. The challenge is to choose from what point the protocol changes from proactive to reactive.

In the following paragraphs, we briefly survey some of the most relevant routing protocols for WMNs.

OLSR The Optimized Link State Routing[10] is a proactive link state protocol for mobile ad hoc networks. It includes a number of optimizations that aim at reducing the cost of forwarding information in the network. In particular, for each node, a subset of neighbors, called the multipoint relays (MPR), are elected to forward announcements. The key idea behind the multipoint relays is to reduce the duplicate retransmissions in the same region.

Algorithm: each node selects its multipoint relay set among its one-hop neighbors in order to cover all two-hop neighbor nodes. Having a bidirectional link towards each of those neighbors is imposed by OLSR. Only MPR nodes are used as intermediate nodes in a route. Each node in the network periodically broadcasts information about its one-hop neighbors which have selected it as a MPR. Upon reception of this MPR selectors list, each node calculates or updates its routes.

The route is then a sequence of hops though MRPs. In order to detect bidirectional links with neighbors, each node periodically broadcasts HELLO messages, containing a neighbor list and their link status. HELLO messages contain the list of addresses of the neighbors to whom the node has bidirectional connectivity and the list of neighbors that are heard by the node. The contents of these messages allow each node to know the existence of neighbors up to two hops and the selection of its MPRs, which are also indicated in the HELLO message. With information extracted from HELLO messages, each node can construct its MPR Selector table.

Each node broadcasts specific control messages called Topology Control (TC), in order to build the routing table for forwarding purposes. TC messages are sent periodically by nodes to declare its MPR Selector set (empty MPR Selector sets are not sent). TC messages are used to maintain topology tables for each node.

B.A.T.M.A.N The Better Approach To Mobile Ad Hoc Networks[11] is another proactive protocol for establishing multi-hop routes in mobile ad-hoc networks. Each node only maintains information about the best next hop towards all other nodes, which avoids unnecessary knowledge about the global topology and reduces the signaling overhead.

Algorithm: each node n broadcasts originator messages (OGM) to inform neighbor nodes about its existence. The neighbors rebroadcast the OGMs to inform their neighbors about the existence of node n, and so on. The network is therefore flooded with these small packets that contain the address of the original node, the address of the node rebroadcasting the packet, a TTL and a sequence number. Each node rebroadcasts the OGM at most once and only if it is received by the current best next hop towards the original initiator of the OGM. Thus

OMGs are selectively flooded through the mesh network. Route discovery and neighbor selection depend upon the the number and reliability of received OGMs. Sequence numbers are used to perceive the OGM freshness, thus any message received with a lower sequence number than the previous one is dropped. Nodes may alter the TTL of their OGMs to limit the number of hops the message traverses. This is useful for backbone nodes that are deployed only for improved connectivity and coverage purposes. BATMAN outperforms OLSR on almost all performance metrics, due to the simplistic approach. By not collecting more information that it can effectively use, and by only getting information about its neighbors, nodes can compute routes in a more efficient manner. Routing overhead is significantly lower than OLSR, proving that sometimes complex approaches lead to less overall performance.

AODV The Ad hoc On Demand Distance Vector[12] is a reactive protocol that creates and maintains routes only when they are requested. On a given node, the routing table stores only information about the next hop to the desired destination and a sequence number received from the destination, preserving the freshness of the information stored.

Algorithm: on demand, route discovery is done by broadcasting a route request message to the neighbors with the destination and sequence number. Each node that receives that request, increases its hop metric and updates its own table. The destination node upon receiving the message, sends a route reply back to the requesting node.

Reactive protocols like AODV tend to reduce the control traffic messages overhead at the cost of increased latency in finding new routes. AODV is loop free and avoids the counting to infinity problem by clever usage of the sequence numbers in control packets.

SrcRR SrcRR[13] is a reactive protocol based on source routing (similar to the Dynamic Source Routing[14] protocol). The protocol is based on the expected transmission count metric (ETX), a metric incorporates the effects of link loss ratio, asymmetry in the loss ratio between the two directions of each link, and interference among the successive links of a path[15].

Algorithm: every node maintains a link cache, which tracks the ETX values for recently established links. Whenever a change occurs in the link cache, the node runs locally the Dijkstra's weighted shortest-path algorithm to find the current minimum-metric routes to all other nodes. When a node wants to send data to an unknown node, it floods a route request. When a node receives a route request, it appends its own ID, as well as the current ETX metric for the node from which it received the request, and rebroadcasts it. If the received route request is the same but over a different route, the node will only forward it if the route metric surpasses the previous one, ensuring that the requesting node will receive the best route. When a node receives a route request for which it is the target, it sends back a route reply. ETX is retrieved by broadcasting a probe that measures the loss rate from each neighbor.

MeshDV MeshDV[16] is a hybrid protocol that uses proactive route computation for mesh routers (based on the Destination-Sequenced Distance Vector protocol[17]) and on-demand path request for mesh clients. MeshDV is an IPv6 only protocol, and uses information from the data link layer to perform route selection.

Algorithm: whenever a node wishes to establish a connection with other node, it sends a neighbor solicitation packet to its mesh router. The mesh router then (if the node is not local) sends a multicast packet to ask other mesh routers for the destination node. The remote mesh router which to the destination node is associated sends back a unicast reply back and the mesh router can now send a neighbor advertisement back to the node requesting the connection. Since routes between mesh routers are proactively maintained, nodes along the path are able to route the new packet (by encapsulation) without knowing the destination client address. Only mesh routers to which the clients are associated are aware of the ongoing communication.

3.3 Evaluation of WMNs

There are several different techniques that can be used to evaluate algorithms and protocols for WMNs, namely: theoretical analysis, simulation, emulation, and real-world experiments.

Theoretical analysis uses mathematical models to derive performance metrics such as signaling cost, throughput, latency, etc. Unfortunately, the complexity of most systems makes them difficult to analyze in this manner for most realistic scenarios.

In a simulation, the algorithms are modeled and evaluated in an controlled artificial environment. This ensures that the evaluation is repeatable and that the user has a tight control on all the parameters that affect the results. Furthermore, it allows to experiment with very large topologies in a cost-effective manner. However there is also a downside to simulations: the lack of realism, since all effects must be simulated, there are many external factors that not considered by the model (interference, reflection, etc), and the results may not be representative of the algorithm behavior in a real-world scenario. Most results are qualitative due to the reasons explained above.

In an emulation, both hardware and software are designed to run under controllable laboratory conditions. An emulator provides a translation layer (usually done by software) from the emulated computer to the computer it is running on. Network emulation is accomplished by introducing a device that mimics the behavior of the environment being emulated. This device may be a computer that incorporates a variety of network attributes into the emulation model such as: RTT (Round Trip Time), available bandwidth, packet loss, duplication of packets, and packet reordering. The advantages of using emulation are repeatability, control over the environment, and a certain degree of realism that the laboratory provides. The costs per test are higher than with simulation but there are scalability bounds to it.

With real-world experiments, all parts of the system are tested under the same operational conditions for which it has been designed to operate. Thus, this approach limits the possibilities of making erroneous or inaccurate assumptions about the impact of external factors. Real-world experiments provide more feedback than simulations or emulations. Furthermore, they are the best way to show that the tested system indeed works as intended. However these experiments lack the repeatability and their scalability is limited due to hardware costs and deployment manpower requirements[18].

3.4 Testbeds

A testbed is a framework which supports testing, comparison and evaluation of algorithms and protocols in the real world. Below we refer to some examples of testbeds that have been deployed to study WMNs.

Roofnet Roofnet[19,20] is a large scale WMN experiment from MIT that uses about 50 nodes in apartments (few nodes are gateways), scattered to ensure that the longest routes are four hops long. The mesh routers are small mini-itx motherboards with a 802.11b/g card. Roofnet uses SrcRR as the routing protocol. The goal of the project is to provide Internet access to the students - nodes are deployed at their apartments. All nodes are running on the same channel, hence the network is a first generation. Distance, SNR (Signal to Noise Ratio), transmission rate, and the packet loss are measured by the software running on the mesh nodes. Roofnet is a good example of real-world experiment since it is widespread over Cambridge and provides Internet connectivity to about 50 households on a day to day basis. However since Roofnets propagation environment is characterized by its strong Line-of-Sight (LOS) component, it does not model a typical WMN since it does not account for obstacles and NLOS environments. Some academic testbeds model this behavior precisely because they are deployed inside University buildings.

UCLA Testbed In the University of California, Los Angeles a testbed comprised of one gateway, four mesh routers (only one of them provides wireless access to clients), and a variable number of clients has been deployed[6]. The nodes are laptops, which increase the cost of deployment, and act as clients or mesh routers that communicate on the same channel (this is a first generation network). The gateway is connected to a FTP server and a streaming server, that are used for testing purposes. The technology used to build the WMN is the Mesh Connectivity Layer (MCL) from Microsoft, an open source tool that implements a modified version of the Dynamic Source Routing (DSR) protocol. The tests were done focusing on the performance of multimedia applications. Metrics were obtained when a flux of data traversed the nodes, such as the packet delivery ratio per hop and the latency caused by the number of hops between source and destination. Some experiments were also done to test the

Quality of Service (QoS) provided by the 802.11e for wireless networks (parameters such as contention window values and inter-frame space number can be altered to differentiate service flows). The testbed lacks the ability to run more routing protocols and is too small to correctly test scalability, since the tests were done solely for the deployed network architecture.

MeshDV Testbed The MeshDV testbed[21] was deployed in LIP6 laboratory of Universite Paris VI. The network is comprised of 12 mesh nodes and is built using Soekris net4521 boxes, running NetBSD and using exclusively the IPv6 protocol stack. There are two wireless interfaces on each node, one for the client sub-network and one for the mesh backhaul, making the network a second generation in terms of configuration. Tests were conducted to measure the hop count impact over traffic, as well as delay and throughput of clients.

UMIC-Mesh The UMIC-Mesh[7] (RWTH Aachen University) is an alternative approach to study WMNs and it is characterized by a hybrid architecture, consisting of real and virtualized testbed. The virtual environment is used for development and validation of functionality. On the other hand, the real testbed is used for execution and evaluation purposes providing a high degree of realism that is needed for this step. The WMN consists of 21 mesh routers in one building and 12 in another, 2 routers are used to interconnect both buildings. Each mesh router is equipped with two 802.11a/b/g NICs: one is used for router-to-client communication, another is used for mesh backhaul, creating effectively a second generation configuration. The testbed provides the option to run only two protocols: DYMO and OLSR (reactive and proactive routing). The chosen metrics for evaluation protocols were: throughput, average hop count, and average packet loss. Tests were done with the purpose of showing that erroneous use of routing metrics (ETX, OLSR HELLO and TC) for wireless multi-hop networks can significantly reduce performance.

The above examples collect common evaluation metrics and all of them designed their own test and reporting tool, they also lack an automatic tool for generating traffic. Evolution of WMN technology depends on the obtained results and laboratory environments can possibly catalyze the proliferation of such technology.

3.5 Monitoring

The purpose of network monitoring is to extract information about the system current configuration, the current values of relevant performance metrics, to detect abnormal or faulty behavior, and forecast potential performance degradation scenarios.

The information obtained via network monitoring can be used by system administrators to solve existing problems, plan the maintenance and future upgrades of the system, etc. The monitoring information may also be used by the protocols that run in the WMNs to optimize their own performance.

Examples of Monitored Values Examples of configuration parameters and performance metrics that can be extracted from a network node are: CPU and memory usage; Uptime; RSSI and Noise (Received Signal Strength Indication and Noise can be used to assess the medium quality); Bit rate; Wireless Channel; MAC Address and IP Address; Clients associated; MTU (the Maximum Transmission Unit is useful to know at which size the packet will fragment); TX, RX, FW packets; TX, RX, FW errors; TX, RX, FW traffic (Transmitted, Received, and Forwarded packet information is useful to understand how the network load is distributed); and Default Gateway.

Some examples of performance metrics that can be used by the routing protocols during their operation are[22]: the Expected Transmission Count (ETX); the Expected Transmission Time (ETT), that also considers link quality by analyzing the time a data packet needs to be successfully transmitted to each neighbor; the Effective Number of Transmissions (ENT), an extension of ETX which considers different link routes or capacities; the Weighted Cumulative ETT (WCETT), that accounts for the interference among links that operate on the same channel, it favors channel diversity and low intra-flow interference.

Monitoring Steps Network monitoring consists of two main steps[23]: measurement phase and gathering phase. In the measurement phase, the state and performance of the nodes is evaluated, while in the gathering phase, the data is collected and analyzed with the purpose of inferring the overall network state. These phases can be implemented using different approaches.

Measurement Phase The measurement phase can be passive or active. Passive measurement consists of capturing and examining individual packets passing through the node, whereas active measurement involves the injection of probe packets into the network. Active and passive measurement approaches have distinct advantages and drawbacks. Active monitoring causes application and measurement traffic competition, while passive monitoring avoids the contention problem. On the other hand, active monitoring improves fault tolerance, provides more up-to-date data and, more importantly, is application or protocol independent, in a sense that passive monitoring is tightly coupled with application or protocol specifications. Both approaches can be combined in a hybrid manner[24]: when the network is saturated traffic-wise, a passive monitoring is used; when the network is on a non-traffic-intensive state, active measurements could be done without compromising the bandwidth offered to clients.

The active measurement phase can also be based on broadcast or unicast traffic. On broadcast-based measurements, each node broadcasts probes to all neighbors at an average period, introducing extra overhead as explained above. On the other hand, unicast-based measurements make use of the real unicast traffic as the natural probing packets without incurring extra overhead. Naturally, this approach only provides monitoring data when there is traffic being routed through the nodes.

Gathering Phase In turn, the gathering phase can be reactive or proactive. With reactive monitoring, the system gathers information only when it is requested (on-demand basis). The event driven monitoring is a particular category of reactive monitoring: data is only transmitted when a determined event occurs. To that end, a threshold-based monitoring is used. However, reactive monitoring does not provide the ability to predict future problems. With proactive monitoring, the systems actively collect and analyze the network on a regular basis to detect faults and predict potential states that compromise network performance. This is especially important if the network status snapshot has to be as up-to-date as possible due to time-critical traffic analysis. However, this approach suffers from high monitoring overheads. The reporting frequency should be selected appropriately, so as to not impact the desired functionality. Due to their complementary nature, proactive and reactive monitoring should not be seen as competitive[25].

The gathering phase can also be classified in two categories, concerning who is in charge of collecting the measures, namely it can be centralized or distributed. On a centralized network monitoring approach, a unique data point collector gathers all the information regarding the network state from a set of agents that are limited to perform only data measurements. This concentration of data processing and analysis on a single node hinders the scalability of the system. In contrast, distributed network monitoring systems are comprised of a hierarchy of top and mid-level managers and bottom-level monitoring agents. Such top-down approach improves scalability and may be further enhanced by developing cooperation protocols between nodes located at the same hierarchy level.

The in the following paragraphs we address some relevant monitoring protocols and systems.

SNMP The Simple Network Management Protocol (SNMP)[26] is the de-facto protocol for management of most networks. SNMP was originally designed for static wired networks and uses a centralized approach for monitoring purposes. The SNMP architectural model is a set of devices that run SNMP agents, which collect local information as defined in the SNMP Management Information Base (MIB). MIBs contain the state of each node in the form of counters and variables.

Network management stations execute management applications which monitor and control network elements. Network elements are devices such as hosts, gateways, terminal servers, and the like, which have management agents responsible for performing the network management functions requested by the network management stations.

SNMP allows for periodic polling of variables in the MIB of each node as well as *traps*, which are triggered by events in the network. A network management system would periodically poll each node from a single location and provide a full network view to its administrator. SNMP traps could also be configured in the agents to respond to local changes. SNMP-based systems can either be proactive or reactive and are typically centralized, having high overheads due to periodic polling of MIBs.

Since SNMP uses a centralized model, its use to monitor WMNs is limited due to the inherent poor scalability. More efficient data exchange and decentralization has to be done in order to consume lesser network resources. One approach to enhance scalability is to limit the amount of monitoring information that is forwarded to the monitoring system.

MeshMon MeshMon is a multi-tiered framework[27] that only monitors a small subset of metrics (baseline metrics) when the network performance is satisfactory. The indication of a potential problem is perceived when those metrics cross a determined threshold, causing the system to transition to collect more detailed metrics. The biggest challenge in designing multi-tiered metric systems is to identify which metrics are strictly necessary to make decisions at each tier and which should be their threshold. When a baseline metric crosses its threshold, the node attempts to locally diagnose the problem, if unsuccessful, it contacts the gateway that will attempt to generate a diagnosis on every node of the node's upstream path. Overhead reduction is then achieved by only transmitting the necessary metrics for a specific problem set.

Scuba Scuba[28] has an approach to monitoring similar to MeshMon, in the sense that limits the amount of observed metrics. It provides a focus and context visualization framework, in which the performance metrics are placed into several tiers or contexts. The topmost context provides the network administrator a holistic overview of the mesh network. This view can be narrowed to focus on the problem region, zooming the level of detail in order to reveal the underlying metrics that are not within their normal range. Contexts are divided in three zones: route, link, and client. The route context displays multi-hop routes between mesh routers and gateways and their metrics. The link context reveals the Expected Transmission Count (ETX) on each link. The client context provides metrics to diagnose causes of poor connection quality to mesh clients. SCUBA has a clear disadvantage that is the non-throttling of monitored data, its rate remains constant even under severe network degradation.

Probing Probing or active measurement has also been researched for emergency scenarios[29]. During rescue interventions, it is important that the monitoring tool provides updated network state information without too much overhead. Two techniques are combined to monitor the network: end-to-end probing and bandwidth estimation. End-to-end capacity estimations can be obtained based on per-hop measurements, even under dynamic network conditions. Each node estimates the link capacity to each of its neighbors by sending packet probes. Two back-to-back packets are sent to each neighbor. First a small packet is sent as a trigger, followed by a probe packet with larger size. The time difference between the arrival of the first and second packet is measured and the result is sent back to the sending node. This solution works on top of the OLSR protocol, that transports the measured link capacity, together with link channel information

and disseminates it throughout the network. The link capacity and channel information of each node on a path can be used to identify the bottleneck-link and to make an end-to-end bandwidth estimation of that path. If for each channel on a path, its capacity is calculated, an estimation of end-to-end bandwidth can be made by extracting the minimum capacity value, which is the bottleneck.

MeshFlow MeshFlow[30] is another probe-based solution. Each node sends a special packet that contains a summary of properties of data packets passing through a mesh router. For each hop in the route the packet traverses, more information is added and hence the growth of the packet size can affect scalability. Existing records in the packet can be shortened by functions like average or maximum values, but the detailed information is lost if such aggregations are made. MeshFlow records of each router are then exported to a collector, that constructs an entire view of the network. Probing is not always very accurate, since it ignores certain factors that affect the packet delivery time and path capacity in a WMN, for e.g.: cross talk between wireless interfaces or adjacent channel interference. Furthermore, the system does not scale very well as node density rises.

MMAN MMAN[31] runs on top of OLSR and relies on multiple monitoring stations that collaborate and combine information. A number of these stations are deployed throughout the network and act as passive monitors. This solution requires the stations to be equipped with two radio interfaces: one for listening to the traffic, another to transmit that information out-of-band between stations and the management unit. Albeit not injecting additional traffic, the stations increase deployment cost by requiring an extra radio interface and an extra network to transfer monitoring data.

DAMON DAMON[32] uses an agent-sink architecture for monitoring mobile networks on top of the AODV protocol. Agents in the nodes discover the sinks automatically through periodic beacons (initiated by the sinks). Beacons can also transport agent-instructions that update the nodes and enable the adaptation to new requirements. The proximity to a sink is determined by the hop count carried in the beacon. Agents at the periphery of two or more sinks can receive beacons intermittently; agent association oscillation is prevented by replacing the primary sink only if a predetermined number of beacons are successively received. The system is only scalable if the number of sinks grows with the number of nodes, in order to achieve load balancing.

JANUS JANUS[33] is another distributed framework, running on top of MCL. It uses Pastry[34], a DHT (Distributed Hash Table) peer-to-peer overlay network, to make information available to all nodes in the system. Each node has a unique identifier (ID), which is the hash of its IP address. The routing algorithm works by resolving a single digit at time. At each step, a node forwards the message

to a node whose ID shares with the key a prefix that is at least one digit longer than the prefix that the key shares with the current node. If such node cannot be found, the message is delivered to the node with closest ID. JANUS also uses Scribe on top of Pastry, in order to build a multicast tree for distribution of publish-subscribe events. While peer-to-peer networks do scale well in an Internet paradigm, in a resource constraint environment such as a WMN, the scalability is poor.

ANMP Distributed systems should be designed to scale well according to the underlying structure. The above systems only perform well for a reduced number of nodes. Hierarchical systems should be designed to account for unbalanced distribution of nodes through the structure. Clustering has been proposed as a technique to tackle the monitoring problem in WMNs through an organized hierarchy of clusters that dynamically and autonomously reconfigure the structure as the network topology changes. Nodes form a monitoring overlay that promotes collaboration and adapts itself to the underlying network dynamics. The intermediate levels of the hierarchy produce summaries of the collected data, in order to compress it, before transmitting data to the upper layers. The cluster-head is a special node in the hierarchy that is elected to coordinate and publish information, it also builds and maintains a local network view (aggregation and correlation of data) of its cluster(s) members and outside connections to neighboring cluster-heads.

ANMP[35] is a monitoring solution for ad hoc networks designed as an extension of SNMP. It uses the same structure and protocol for data collection through MIBs. Cluster-heads poll information from their cluster members, which creates unnecessary overhead. The clusters are not dynamic and this solution has not yet been implemented nor tested.

Self-Organized Management Overlay A more recent clustering solution was proposed in [23]. The cluster formation is triggered by the addition of a new node at any time. When a node joins the network it broadcasts a cluster-head query, and its neighbors rebroadcast the message up to j-hops away, being j a configurable parameter. If after a determined time the node does not receive a reply, it promotes itself to cluster-head. A cluster-head may either accept or refuse the new node into its cluster, subjected to different criteria, for example, QoS, load or location. Cluster-heads periodically poll cluster member nodes to verify if they are alive. On the other hand, cluster member nodes expect to be polled, assuming that a cluster-head has disappeared in case the poll messages are not received. The absence of polling messages triggers a cluster-head promotion. Cluster size has to be determined before deploying the nodes, which may raise problems if the network density increases.

Mesh-Mon Mesh-Mon[36] is another clustering solution. It operates according to three principles: each mesh node must monitor itself, each mesh node must

monitor its k-hop neighbors, and each node must help in forming a hierarchical overlay network for propagation of monitoring information. The first principle states that each mesh node must measure its own local information, the second principle aims for a distributed analysis allowing nodes to cooperate and detect local problems, while the third principle is a common approach to achieve scalability through aggregation of data.

Mesh-Mon uses a combination of active and passive monitoring techniques, and a rule-based diagnosis engine. The information collected is concerned with the system configuration and measurements from the physical, link, and network layers. Periodically, nodes probe each other to measure bandwidth and latency among clients, mesh nodes, and external hosts. The measured information is summarized and disseminated to other nodes. To enhance scalability, more information is stored about local neighborhood than about nodes far away.

Mesh-Mon nodes can communicate using flooding if the routing protocol fails or is disabled. For networks with considerable size, the flooding is limited to k-hop neighbors, and thus forming a hierarchical overlay of MeshLeaders, which are responsible for exchanging information between k-hop neighborhoods. A MeshLeader is selected by its k-hop neighbors using a leader-election protocol. Nodes appoint themselves as MeshLeaders, if none exists. Discovering of other MeshLeaders is done through a beaconing process. The importance of a node in a mesh network can be characterized by the number of routing links the node shares with its neighbors. Assuming global topology is available, all the nodes can be ranked according to their degree of importance. A better connectivity rank can be calculated using eigenvector centrality (EVC). EVC is calculated using the network global topology and it is proportional to the sum of the centrality values of all neighboring nodes. A node with a high value of EVC is a strong candidate for MeshLeader. EVC can be calculated using three proposed variants: binary adjacency matrix representing the global topology, ETX, and gateway EVC, in which the importance of Internet gateways is emphasized.

Astrolabe Astrolabe[37] uses a gossip protocol as the method for dissemination queries and results. The key idea behind gossip protocol is simple: periodically, each agent (running on every node) selects other agent at random and exchanges information with it. As time passes, the data will tend to converge (if agents are in different *zones*, then they exchange data associated with their least common ancestor *zone*). Each *zone* elects the subset of agents that gossip on its behalf. The election algorithm can either be arbitrary or deterministic, based on characteristics like load, uptime or even agent coverage (*zones* that the agent represents). When it is time to gossip, the agent picks at random one of the child *zones*, other than its own. Next, the agent looks up for the contacts attribute for the selected *zone* and randomly picks another agent from the set of hosts in the list and proceeds to contact it and send attributes of all child zones at that level and for higher levels up to the root of the tree. The contacted agent compares the information received with his own and updates his out-of-date information and sends its own information back to the gossiper. Astrolabe adopts a weak

System	Gathering	Structure	Implemented	Routing protocol dependent	Scalable
MMAN	proactive (off-band)	plain	yes	yes (AODV)	no
DAMON	proactive	hierarchical	yes	yes (AODV)	yes
JANUS	hybrid	plain (DHT)	yes	yes (DSR variant)	no
ANMP	hybrid	hierarchical	no	yes	yes
Cluster	hybrid	hierarchical	yes	yes (OLSR)	yes
Mesh-Mon	proactive	hierarchical	yes	no	yes
Astrolabe	hybrid	hierarchical	yes	yes	yes

Table 1. Comparison between distributed solutions.

notion of consistency, which means that updates will eventually be reflected in every node. Astrolabe also allows the use of SQL queries to search or subscribe to certain events. The queries provide more granularity in monitoring information access than the summarized information exchanged between nodes.

3.6 Summary of the Related Work

Table 1 summarizes the differences between the distributed solutions previously analyzed. In a resource constraint environment such as a WMN, monitoring solutions will tend to be decentralized and distributed, passing some of the intelligence usually in the core of the monitoring system to each node that does not play a passive role anymore. The choice between proactive or reactive measurement depends on the resource or variable that requires monitoring, if one needs an historical chart of such resource, then a proactive approach must be taken, if the report can be done via events that only alert if something is wrong, then a reactive measurement can be chosen. Typical clustering systems have passive measurement capabilities and adopt a decentralized architecture to distribute the task of monitoring the network between their nodes, which gather data in a proactive or reactive manner.

The main challenges faced when developing a monitoring system are: minimize bandwidth consumption, minimize the size of monitoring information while providing important information for diagnostic of network health, adaptation of monitoring systems to underlying network conditions, resilience to cluster-head or gateway (sink) failure, minimize the resources consumed by mesh routers (CPU and memory), automatic generation of traffic for protocol evaluation and validation. All these challenges have to be faced while maintaining an up-to-date information of network conditions. An approach to WMN monitoring and testing that addresses these challenges is presented in the next section.

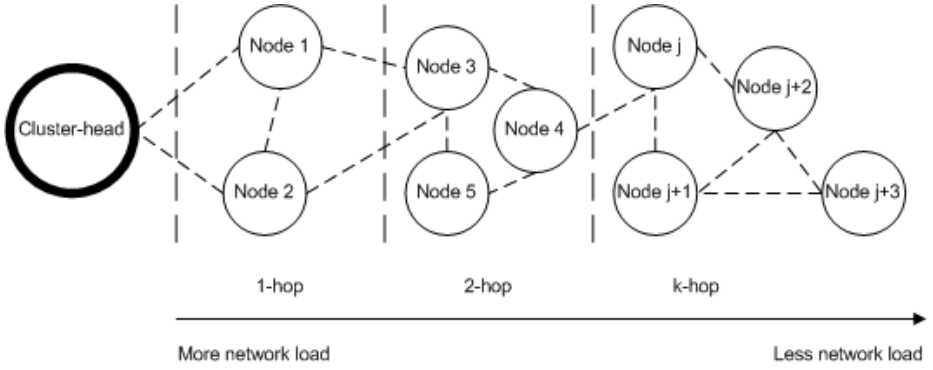

Figure 3. Proposed architecture for the WMN monitoring system.

4 Architecture

4.1 Overview

As discussed in the previous section, the overhead that a monitoring system induces in a WMN may interfere with the normal operation of the network and have an impact on the quality of service perceived by the clients. Clustering-based solutions are an efficient and scalable approach, that allows to minimize the impact of monitoring activity in the overall network performance; however, the proposed solutions create static clusters that do not adapt to the underlying network conditions. Also, the surveyed solutions are tied to a given routing protocol. So, a monitoring solution that is independent of the routing protocol has several advantages: it can be used in several systems, regardless of the routing protocol in use, and it can monitor the performance of routing protocol itself.

The proposed architecture is a refinement of a cluster-based system that adapts its structure in order to minimize the interference with the data traffic. The architecture is depicted in Figure 3. Every node in the WMN sends its monitoring data, using the Layer 2 protocol, to a cluster-head. Cluster-heads are responsible for aggregating the monitoring information and for sending the result to the closest gateway. The cluster size is a function of the current resource consumption in the region: if network load increases, the cluster size decreases and vice-versa. With this mechanism we aim at lowering the interference between monitoring and client traffic. The gateways use the fixed infrastructure to route the monitoring information to a central network management server.

We will also develop traffic generators that may be activated in any desired node of the system. Traffic generators help when profiling the system and may even help the system diagnosis. Remote configuration and activation of traffic generators will be supported.

4.2 Minimizing The Monitoring Overhead

In our approach, we will attempt to minimize the interference that monitoring traffic may have on the applications running on the WMN. Therefore, we will design mechanisms to adapt the amount of information exchanged in function of the observed network utilization. The idea is that more detailed monitoring information may be exchanged when network bandwidth is available, but when the network bandwidth is scarce only summarized information is propagated.

In our approach, nodes periodically send monitoring information to their cluster-heads. We will attempt to piggyback control messages whenever possible, to minimize the number of packets transmitted over the network. For instance, nodes that forward control messages from other nodes to the cluster-head, can piggyback their control information on the forwarded messages.

Cluster-heads may summarize the information collected before reporting to the central monitoring system. Aggregation functions may vary according to the metric being monitored, and should be designed to minimize the information loss. For instance, for a given metric, aggregated data may contain both the average and the extreme values observed in a given region. Examples of metrics that can be aggregated are: RSSI and Noise, Clients associated, and traffic sent.

Detailed information about a node can be obtained by querying its cluster-head, much like Astrolabe[37]. Several levels of granularity (maximum or minimum value, value greater or smaller than x, value equal or different than x) can be achieved by querying a node.

Another step to further reduce message size is to evaluate which metrics have lower probability of changing its value. For example: the wireless channel is typically fixed for every node, and it only makes sense reporting it when there is a change. Reactivity can also be configured in nodes if a metric only makes sense reporting after it crosses a determined threshold.

Since cluster-heads aggregate monitoring information collected by several nodes, even the size of summarized information can be large. Therefore, the summarized information may be fragmented in several packets to minimize the interference of the signaling traffic on on-going data flows, in particular on latency-sensitive streams that use the same paths as the ones used to send control information to the gateways.

4.3 Routing of Monitoring Information

The monitoring system needs to route the monitoring information to the gateways. In a first step, nodes route their data to cluster-heads. Then cluster-heads route summarized information to the gateways. These routes must be computed in a manner that is independent of the routing mechanisms being in use in the WMN, given that we aim at a solution that is protocol independent.

A simple and pragmatic process to discover such paths is to use an approach based on the mechanisms that have been proposed for BATMAN[11]. Gateways and cluster-heads periodically send beacon messages that every node rebroadcasts. Nodes select the best next hop towards a gateway or to the cluster-head

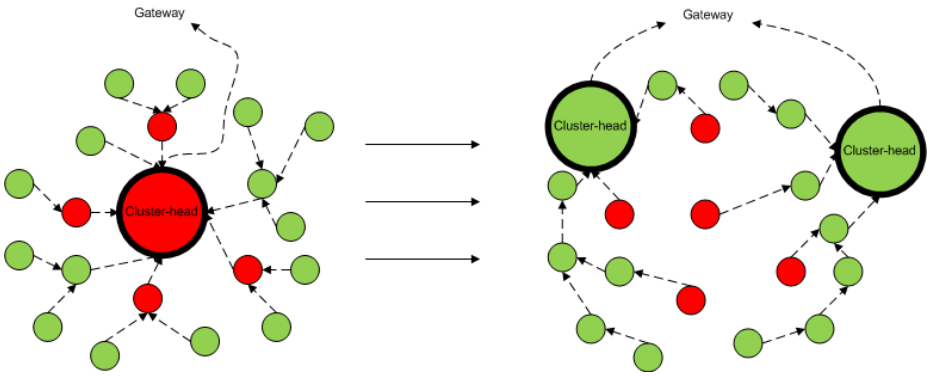

Figure 4. Clustering Offload system. *(note: a red node represents a node under heavy network load, while a green node represents a node that has lower or no traffic passing through.)*

according to the number of successful beacon messages received by its neighbors. The presence of multiple gateways and cluster-heads cause unnecessary beacon messages that will not be used by nodes far away from them. To limit the flooding of such messages, every node receiving more than one beacon message will only rebroadcast it if the received message is with higher quality than the other beacon message, causing messages with less quality to be dropped. This causes a border around cluster-heads and gateways and ensures that nodes will only receive beacons from the closest ones. Oscillation is prevented by replacing primary cluster-heads or gateways only when a predetermined number of successive beacons is successfully received.

4.4 Adaptive Clustering

Cluster-heads receive monitoring information from all cluster members. In order to minimize the impact on the network performance, the radius of the cluster can be reduced or augmented dynamically. Cluster formation is triggered when a node does not receive beacon messages for a predetermined amount of time. At this moment, it appoints itself as cluster-head and starts the beaconing process. Although this process is dynamic, the cluster-head can be in the center of a zone under heavy network traffic. Since the cluster-head has a holistic view of the zone it can suppress its responsibility and designate another (less loaded) node as cluster-head, allowing the clients to become less affected by signaling traffic. This off-load mechanism is illustrated in Figure 4.

4.5 Traffic Generators

Nodes have the ability to generate traffic (TCP and UDP streams) on demand. This feature is useful to debug routing protocols or simply test the network load

that a certain region can sustain. Besides flow generation, nodes can report the common routing metrics mentioned in Section 3.5.

Automatic traffic generation and test scenarios saves considerable time while developing and deploying a WMN. Tools like Iperf[2] and ping can be used to either generate traffic or measure latency between nodes.

5 Methodology for Evaluating the Work

To evaluate the monitoring algorithms and the prototype implementation we will use a combination of simulations and experiments on a real testbed.

Simulations will be used to capture performance metrics of the monitoring algorithms under different scenarios. The relevant metrics will be the amount of signaling data generated and the latency of the monitoring mechanisms. Typical scenarios of wireless usage will be tested with the use of traffic generating tools. In addition, the impact of monitoring data transmissions in multimedia streams, like VoIP will be evaluated. The adaptive clustering algorithm will be tested by injecting traffic near the cluster-head and verifying the breath effect as well as its relocation (if the zone is under heavy network load) to border areas that are under lower network load. A network simulator, such as NS-2[3], will be used for this purpose, given that this simplifies the process of assessing the performance of the algorithms in WMNs of different sizes and topologies.

The simulations will be complemented by a real deployment using a mesh based on FON routers that will be configured for the effect, using OLSR or BATMAN as routing protocols. The FON router is a device with the following technical specifications:

- dimensions: 93.5 mm x 25.5 mm x 110 mm
- 1 IEEE 802.11b / 802.11g interface
- 1 RP-SMA connector (reverse SMA)
- 1 detachable antenna (1,5 dBi)
- 2 ethernet ports 10/100Mbps (1 WAN + 1 LAN)

On this experimental setting, we will compare the performance of our prototype against a simple solution based on SNMP, configured with proactive monitoring.

6 Scheduling of Future Work

Future work is scheduled as follows:

- January 9 - March 29: Detailed design and implementation of the proposed architecture, including preliminary tests.
- March 30 - May 3: Perform the complete experimental evaluation of the results.
- May 4 - May 23: Write a paper describing the project.
- May 24 - June 15: Finish the writing of the dissertation.
- June 15: Deliver the MSc dissertation.

[2] http://sourceforge.net/projects/iperf/
[3] http://www.isi.edu/nsnam/ns/

7 Conclusion

This report addressed the problem of implementing a set of monitoring tools for Wireless Mesh Networks. After a brief survey on this technology, we have listed the main approaches to perform monitoring in this context and identified their limitations. We have sketched an alternative solution that aims at reducing the overhead induced by the monitoring activities. We plan to implement this system and evaluate it using both simulations and on an experimental testbed.

Acknowledgments I am grateful to J. Mocito, D. Monica, and T. Taveira for the fruitful discussions and comments during the preparation of this report. This work was partially supported by project "Redico" (PTDC/EIA/71752/2006).

References

1. Bing, B.: Emerging Technologies in Wireless LANs: Theory, Design, and Deployment. Cambridge University Press, New York, NY, USA (2007)
2. Jubin, J., Tornow, J.: The darpa packet radio network protocols. Proceedings of the IEEE **75**(1) (Jan. 1987) 21–32
3. Wang, X., Lim, A.: Ieee 802.11s wireless mesh networks: Framework and challenges. Ad Hoc Networks **6**(6) (2008) 970–984
4. Hiertz, G., Max, S., Zhao, R., Denteneer, D., Berlemann, L.: Principles of ieee 802.11s. In: Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on. (Aug. 2007) 1002–1007
5. Akyildiz, I., Wang, X., Wang, W.: Wireless mesh networks: a survey. Computer Networks ISDN Systems **47**(4) (2005) 445–487
6. Hamidian, A., Palazzi, C., Chong, T., Navarro, J., Korner, U., Gerla, M.: Deployment and evaluation of a wireless mesh network. Advances in Mesh Networks (2009)
7. Zimmermann, A., Schaffrath, D., Wenig, M., Hannemann, A., Gunes, M., Makram, S.: Performance evaluation of a hybrid testbed for wireless mesh networks. In: Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on. (Oct. 2007) 1–10
8. Kiess, W., Mauve, M.: A survey on real-world implementations of mobile ad-hoc networks. Ad Hoc Networks **5**(3) (2007) 324–339
9. Abolhasan, M., Wysocki, T., Dutkiewicz, E.: A review of routing protocols for mobile ad hoc networks. Ad Hoc Networks **2**(1) (January 2004) 1–22
10. Jacquet, P., Mhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., Viennot, L.: Optimized link state routing protocol for ad hoc networks. In: Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International. (2001) 62–68
11. Johnson, D., Ntlatlapa, N., Aichele, C.: A simple pragmatic approach to mesh routing using batman. In: 2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, CSIR, Pretoria, South Africa. (2008) 10
12. Huhtonen, A.: Comparing aodv and olsr routing protocols (2004)
13. Aguayo, D., Bicket, J., Morris, R.: Srcrr: A high throughput routing protocol for 802.11 mesh networks. Technical report (2003)

14. Johnson, D., Maltz, D., Broch, J.: Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In: In Ad Hoc Networking, Addison-Wesley (2001) 139–172
15. Couto, D.D., Aguayo, D., Bicket, J., Morris, R.: A high-throughput path metric for multi-hop wireless routing. Wireless Networks **11**(4) (2005) 419–434
16. Iannone, L., Fdida, S.: Meshdv: A distance vector mobility-tolerant routing protocol for wireless mesh networks. IEEE ICPS Workshop on Multi-hop Ad hoc Networks (2005)
17. He, G.: Destination-sequenced distance vector (dsdv) protocol. Technical report, Networking Laboratory. Helsinki University of Technology
18. Kiess, W., Mauve, M.: A survey on real-world implementations of mobile ad-hoc networks. Ad Hoc Networks **5**(3) (2007) 324–339
19. Aguayo, D., Bicket, J., Biswas, S., Couto, D., Morris, R.: Mit roofnet implementation
20. Bicket, J., Aguayo, D., Biswas, S., Morris, R.: Architecture and evaluation of an unplanned 802.11b mesh network. In: MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking, New York, NY, USA, ACM (2005) 31–42
21. Iannone, L., Kabassanov, K., Fdida, S.: The meshdvnet wireless mesh network test-bed. In: WiNTECH '06: Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization, New York, NY, USA, ACM (2006) 107–108
22. Campista, M., Esposito, P., Moraes, I., Costa, L., Duarte, O., Passos, D., Albuquerque, C., Saade, D., Rubinstein, M., Rubinstein, M.: Routing metrics and protocols for wireless mesh networks. Network, IEEE **22**(1) (Jan.-Feb. 2008) 6–12
23. Sailhan, F., Fallon, L., Quinn, K., Farrell, P., Collins, S., Parker, D., Ghamri-Doudane, S., Huang, Y.: Wireless mesh network monitoring: Design, implementation and experiments. In: Globecom Workshops, 2007 IEEE. (Nov. 2007) 1–6
24. Lowekamp, B.: Combining active and passive network measurements to build scalable monitoring systems on the grid. SIGMETRICS Perform. Eval. Rev. **30**(4) (2003) 19–26
25. Gupta, D., Wu, D., Mohapatra, P., Chuah, C.: A study of overheads and accuracy for efficient monitoring of wireless mesh networks. Pervasive and Mobile Computing (2009)
26. Schoffstall, M., Fedor, M., Davin, J., Case, J.: Simple network management protocol (snmp) (1990)
27. Raghavendra, R., Acharya, P., Belding, E., Almeroth, K.: Meshmon: a multi-tiered framework for wireless mesh networkmonitoring. In: MobiHoc S3 '09: Proceedings of the 2009 MobiHoc S3 workshop on MobiHoc S3, New York, NY, USA, ACM (2009) 45–48
28. Suwannatat, A., Kevin, T., Almeroth, C.: Scuba: Focus and context for real-time mesh network health diagnosis. In: PAM. Volume 4979 of Lecture Notes in Computer Science., Springer (2008) 162–171
29. Naudts, D., Bouckaert, S., Bergs, J., Schoutteet, A., Blondia, C., Moerman, I., Demeester, P.: A wireless mesh monitoring and planning tool for emergency services. In: Proc. The 5th IEEE Workshop on End-to-End Monitoring Techniques and Services. (May 2007)
30. Huang, F., Yang, Y., He, L.: A flow-based network monitoring framework for wireless mesh networks. Wireless Communications, IEEE (2007)

31. Kazemi, H., Hadjichristofi, G., DaSilva, L.A.: Mman - a monitor for mobile ad hoc networks: design, implementation, and experimental evaluation. In: WiNTECH '08: Proceedings of the third ACM international workshop on Wireless network testbeds, experimental evaluation and characterization, New York, NY, USA, ACM (2008)
32. Ramach, K.N., Belding-royer, E.M., Almeroth, K.C.: Damon: A distributed architecture for monitoring multi-hop mobile networks. In: In Proceedings of IEEE SECON. (2004)
33. Scalabrino, N., Riggio, R., Miorandi, D., Chlamtac, I.: Janus: A framework for distributed management of wireless mesh networks. In: Proceedings of the 3rd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities. (2007)
34. Rowstron, A., Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: In Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware). (2001) 329–350
35. Chen, W., Jain, N., Singh, S.: Anmp: ad hoc network management protocol. IEEE Journal on Selected Areas in Communications (1999)
36. Nanda, S., Kotz, D.: Mesh-mon: A multi-radio mesh monitoring and management system. Computer Communications **31**(8) (2008) 1588–1601
37. Renesse, R.V., Birman, K., Vogels, W.: Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. ACM Trans. Comput. Syst. **21**(2) (2003) 164–206