MeshAdmin: an Integrated Platform for Wireless Mesh Network Management

Rafael De Tommaso do Valle
Telecommunications Engineering Department
MídiaCom Lab - Fluminense Federal University
Niterói, Brazil
rafael@midiacom.uff.br

Débora Christina Muchaluat-Saade
Computer Science Department
MídiaCom Lab - Fluminense Federal University
Niterói, Brazil
debora@midiacom.uff.br

***Abstract*—Wireless mesh network management is more complex than wired network management mainly because of network resource constraints and link quality variability. Therefore, monitoring solutions for wired networks do not achieve satisfactory performance in mesh networks. There are several techniques and tools proposed in the literature for wireless mesh network management. Each of those techniques and tools usually includes a set of specific functionalities leading to the use of more than one together in order to manage a real mesh network. This article discusses a set of requirements of an integrated platform for wireless mesh network management. This work also presents MeshAdmin, a management platform that fulfills a subset of the presented requirements. In order to evaluate MeshAdmin monitoring traffic overhead, this work presents performance tests done in a real mesh testbed.**

***Index Terms*—wireless mesh networks; network management, network monitoring, resource monitoring**

I. Introduction

Wireless Mesh Networks (WMNs) are self-setup wireless networks that interconnect a group of nodes, usually placed in fixed locations. Mesh nodes are capable of forwarding packets through multihop communication in wireless links until they reach their destination. Nodes usually use the IEEE 802.11 standard in ad hoc mode and build a wireless backhaul for broadband transmission where cable structure is not available and the deployment cost of other technologies is high. Besides the advantage of being low-cost and easy-to-deploy, mesh networks are also fault-tolerant because they provide several paths between packet source and destination [1].

Mesh networks can be used for providing broadband access inside university *campi* and their neighborhood and also for building digital cities, which is a way of enabling digital inclusion in developing countries. Mesh networks require less infrastructure expenses when compared to wired network infrastructure, while they provide connectivity to a much larger area than conventional wireless networks with hotspots.

In order to facilitate the dissemination of this technology, a management platform that facilitates the operation and maintenance of wireless mesh networks is needed.

WMN management is more complex than wired network management mainly because of network resource constraints and link quality variability [2]. Therefore, monitoring solutions for wired networks do not achieve satisfactory performance in mesh networks [3]. In addition, there are significant differences in WMN management when compared to infrastructured wireless network management because of the greater reliability and transmission capacity of wired backbones [4].

Moreover, there are a few available commercial solutions for managing mesh networks and tools are usually developed exclusively for proprietary mesh solutions. Those solutions have high cost, which can prevent their use in projects that have limited resources, such as digital inclusion and digital city projects.

In the literature, there are some techniques and tools developed to assist the management of WMNs (some of them are presented in Section II). Usually, each tool individually is not enough to manage a mesh network itself, requiring the use of more than one together. Using different tools causes information dispersion, since each tool has its own database and/or visualization interface, and some functionalities overlap when different tools deal with common network parameters. It might even be interesting to compare information from different network management tools. However, it means more CPU and memory consumption in wireless routers and more data being transmitted in the network. As wireless routers usually have limited resources and wireless channels have limited capacity, it is desirable to use the least of mesh network resources.

In this context, there is a need for developing an integrated platform for wireless mesh network management, gathering information from wireless routers in one unique database and allowing its manipulation through the same interface. The main goal of this work is to define the requirements of an integrated platform for WMN management and to present MeshAdmin, an implementation of the proposed platform.

The remainder of this paper is organized as follows. Section II discusses several tools and techniques used for WMN management. Section III addresses the main challenges about WMN management and proposes a set of requirements of an integrated platform. Section IV presents MeshAdmin, an integrated platform for wireless mesh network management. Section V evaluates MeshAdmin considering monitoring traffic overhead. Section VI compares MeshAdmin with related work. Finally, Section VII brings final remarks and future work.

978-1-4673-0269-2/12/$31.00 ©2012 IEEE

II. Related Work

There are several techniques and tools proposed in the literature for assisting the management and monitoring of resources and users in wireless networks and, more specifically, in mesh networks.

Sailham et al. [3] discuss WMN monitoring and divide it in two steps: measurement phase, which lies in evaluating the state of network elements along with their performance, and gathering phase, which corresponds to the collection of the measurement data for inferring the state of the overall network. Their work proposes an architecture that organizes the network nodes into a cluster-based hierarchical structure, which is used for delivering monitoring information. It defines cluster-head as "a node that is elected to coordinate and publish information relating to the topology of the cluster(s) and cluster elements that are under its supervision and to the neighbor cluster heads." They identify two basic mechanisms for monitoring mesh networks, the first one is based on SNMP MIB and the second one relies on listening to the OLSR traffic [5], and they choose the second approach. It also proposes a directory service, which aims at archiving collected information about network nodes and map nodes with their corresponding information structure. Moreover, it presents an event notification model based on a publish/subscribe paradigm.

Nanda and Kotz [4] propose a management system for proactive mobile multi-radio mesh networks, designed for dynamic environments, called Mesh-Mon. Its design is based on three main principles: "(i) Each mesh node and client must monitor itself; (ii) each mesh node must monitor its k-hop neighbors and maintain a detailed representation of the local network and a sparse representation of the global network; and (iii) each node must help forming a hierarchical overlay network for propagation of monitoring information". They define Mesh-Mon as the software installed in each mesh node that manages collection, communication and analysis of relevant data, and Mesh-Mon-Ami as a software component that runs in mesh clients and assists communication of management information in disconnected areas.

Huang et al. [6] define as MeshFlow the concept of network traffic flow in mesh networks, based on Cisco international standard NetFlow [7]. The MeshFlow framework is designed to generate, transmit and analyze MeshFlow records in order to perform network monitoring. Each MeshFlow record contains information about data packets transiting in each mesh router. Those records are kept in the node and after a predetermined time are sent to a dedicated server responsible for storing and analyzing collected data.

WiMFlow [8] is a dynamic and self-organized flow monitoring framework in WMNs. Unlike Huang et al. framework, it elects one nodes, in the path of a flow, to monitor it. This aprouch reduces the export overhead traffic. Hence, in a network where topology and routing tables changes often, some flows will not be recorded by this framework.

Jardosh et al. [9] present SCUBA, a tool for real-time monitoring of WMNs. It has three contexts, where each one consists of one or more metrics. The route context shows information regarding throughput and RTT (round-trip time). The link context presents link quality based on ETX (expected transmission count) metric [10]. And the client context provides the number of clients associated with each router, the percentage channel utilization per client, the received signal strength indicator (RSSI) of MAC-layer frames received from clients and the amount of external interference.

Abaré [11] is a coordinated and autonomous framework for deployment and management of WMNs. It has 3 layers: the admin layer responsible for interaction with the user; the core layer, which represents the core system where the logical part and storage of information are located; and the router layer, which allows access to the router for direct communication with the operating system of each node.

Jan et al. [12] propose a monitoring tool for multi-hop network. Their tool provides a Java GUI for network administrators and an API that allows remote access to the monitoring information. In their approach, each node has its own trace file that gathers information about the network topology from a modified version of its radio driver. It also stores other relevant information about the node, as CPU info, memory usage and network interface traffic. That trace file is sent through the network via FTP to a collector every second and stored in a MySQL database server. Its GUI provides to the network administrator global information about the network, network topology, node information and alert messages.

Maya [13] is a tool for WMNs management. Its architecture is based on the cooperation of three different components: the management unit, the distribution network, and the network clients. The management unit allows the network administrators to manage and configure a set of nodes through a simple graphic interface with only one action.

Mesh Topology Viewer (MTV) [14] aims at showing real-time mesh network topology and quality of its links. It shows link quality based on ML (minimum loss) metric value provided by OLSR-ML routing protocol [15]. It generates an SVG file with a background image (e.g. map, plant) defined by the network administrator, presenting mesh nodes, and shows colored network links according to their quality.

WiFiDog [16] is a captive portal solution that provides user authentication and centralized monitoring of an infrastructured wireless network. CoovaChilli [17] is another tool used for the same purpose. It is a captive portal tool for user access control, which provides authentication via RADIUS protocol. Both solutions also provide user statistics about network use.

III. Integrated Platform for WMN Management

As mentioned before, WMN management is a more complex task than management of wired or infrastructured wireless networks [2]–[4]. The main issues about management of mesh networks are resource-constrained devices, as available disk and memory; wireless backbone with limited capacity; large variability of wireless link quality; and unfriendly node placement, such as on top of buildings, towers, lamp posts etc.

A more detailed discussion about issues in WMN management can be found in [2].

In general, the features of a network management platform include collecting and storing statistics about nodes and links; network monitoring, providing network topology visualization, indicating wireless link quality and facilitating the discovery of network failures; and user access control.

Based on related work and our previous experience with mesh network, we define a set of functional requirements of an integrated WMN platform. These functional requirements are detailed in the following subsections.

A. Data Collection

A management platform should be able to collect data related to mesh nodes and links. Analyzing those data, it is possible to observe the network's overall behavior. With the assistance of monitoring techniques, it is possible to identify and repair failures.

Each node has a set of parameters that need to be observed in order to identify failures in its operation, such as uptime, CPU usage, available/used memory, bytes in and out in LAN, WLAN and WAN (gateways) interfaces, radio information (such as transmission rate, transmission power etc) and even other specific data, such as information about its power system.

Link quality is also an important parameter to be collected, since it is possible to observe link problems and identify inaccessible nodes. In the literature, there are several references to WMN metrics developed to measure link quality [1].

B. Data Storage

A management platform should also keep collected data for a specified amount of time. It is not a good practice to store those data in routers. Mainly because of their resource limitations and, moreover, in case of a node becoming inaccessible, it would not be possible to access the information stored in that node. Thus, there should be an external database server to store all information gathered by the WMN platform.

However, for a short period, it may be necessary to store some information in mesh nodes, so they can be sent to the external database later. Therefore, each node should retain some memory space to store data. The storage period in nodes must be defined considering available memory space and overhead traffic generated by management traffic. An intelligent mechanism, as proposed in [6], should be implemented for this purpose.

C. Monitoring

Collected data should be monitored continuously to make possible the identification and solution of network failures. Monitoring tools should be designed so the network administrator can obtain network diagnosis more accurately.

An essential feature for a management platform in terms of monitoring is real-time topology visualization [9], [14]. A platform should be able to inform the network administrator about wireless link quality – according to a convenient metric – so he can identify failures. It is advisable that the topology view provides geographical position of nodes, making easier the identification of each mesh node location.

Another key feature is the possibility of visualizing collected information from network elements. In case of any abnormality, the administrator should be able to query historical series of data stored in a database and identify failure patterns, e.g. abrupt decrease in available memory, lack of disk space, high rates of discarded packets in an interface etc.

In addition, the platform should be able to send alerts to the administrator about network health, which we defined as active monitoring. It should observe collected data and compare with predetermined thresholds and/or data previously gathered. In case of abnormality, the platform could send the administrator an alert in different manners, e.g. email, SMS, twitter etc.

Monitoring traffic should not compete with user traffic. The network should have different service classes for management and user traffic in order to avoid management traffic from compromising user traffic.

D. User Control

It is expected that a WMN management tool controls user access. Moreover, it should provide information about registered users and their access profiles. This information can be divided in two groups, user connection information and user flow information.

Considering user connection, it is desirable to provide: a list of connected users, the mesh access point used for connection, the user IP address, the number of user flows, the timestamp of the beginning of user connection, the timestamp of the end of user connection (if finished) and the bytes downloaded and uploaded in each connection.

User flow information should also be collected, such as: source port, destination IP address, destination port, transport protocol, number of packets sent, number of packets received, bytes sent by each flow, bytes received by each flow and gateway used by each flow.

Flow analysis is also important from the security perspective. It is possible to identify malicious traffic and from which user that traffic came. Moreover, with flow information the administrator can identify saturation points, i.e., access points that are being accessed by a considerable number of users, and redesign the network, relocating or adding mesh nodes in order to balance network traffic [6].

Besides these requirements that are met by most of user control tools for infrastructured wireless networks, there are some specific requirements for WMNs. They are described as follows.

Multihop Support: Communication in mesh network takes place through multiple hops. When a user registers himself in the network, he should be able to authenticate from any access point.

Multiple Gateway Support: WMNs can have one or more gateways (node that connects the local backbone to the Internet) [18]. This implies that an authentication tool should have an integrated user database, which can be queried by all gateways at the time of authentication. Thus a user can register through any mesh gateway and will be able to authenticate through any other.

Multiple Interface Authentication: A mesh network advantage is the ability of client connection through wired or wireless interface. Therefore, a user control tool should be able to authenticate users in both interfaces.

IV. MeshAdmin Platform

Based on the requirements previously discussed, this paper proposes MeshAdmin, a management platform for wireless mesh networks, which use low-cost and easy-to-find wireless routers, such as Linksys WRT54g and Ubiquiti Bullet. That kind of routers has resource constraints in terms of RAM, flash memory and CPU processing. They run a firmware based on the OpenWrt Linux distribution. Wireless routers can also be placed where there is no conventional power source, so they may be powered by batteries, which are charged by solar panels.

MeshAdmin current implementation is divided in the following modules: Admin Panel, Collection Module, Storage Module, Alert Module and Display Module. Figure 1 shows MeshAdmin modules interaction.

The Admin Panel inserts information about networks that will be monitored in the Storage Module and sets some parameters of the Collection Module, as interfaces and disks that will be monitored in each node. The Storage Module also receives data from the Collection Module - monitoring data from links and nodes, and from the Alert Module. The Display Module queries the Storage Module in order to obtain information that will be displayed. For real-time monitoring, the Display Module also receives data from the Collection and Alert Modules. Each one of those modules are described as follows.

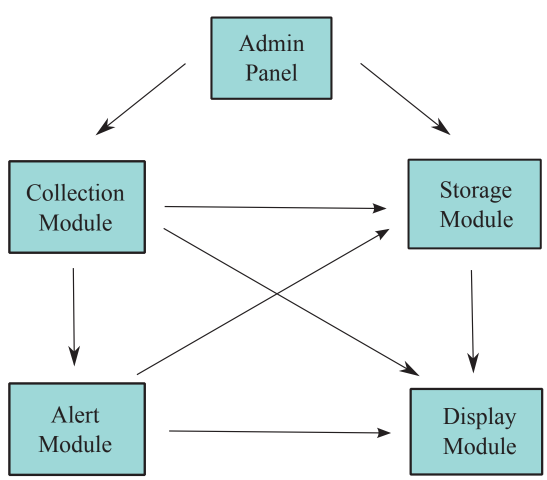

Fig. 1. MeshAdmin operation scheme.

A. Admin Panel

MeshAdmin Admin Panel is used by the network administrator for defining mesh networks that will be monitored, network nodes and some network and node parameters. It is divided in four configuration groups:

- *Auth* for user and group creation and its permissions;
- *Configuration* for network and node addition and gateway definition;
- *Diagnosis* for defining disks and interfaces that will be monitored;
- *Monitoring* for configuring the routing metric used by the routing protocol.

B. Collection Module

The Collection Module is responsible for collecting monitoring data from nodes and links.

1) Node Data Collection: In order to collect information about mesh nodes, the MeshAdmin server has a network management system that communicates with agents installed in each router using the SNMP protocol. That agent is a compact implementation of SNMP and was chosen not to compromise the use of router memory and CPU.

Two SNMP implementations developed for OpenWrt were tested, Net-SNMP[1] and Mini SNMP Daemon[2]. The first spends a high memory percentage, while the second performed well and did not compromise any router resource. Therefore, Mini SNMP Daemon is used as the router SNMP agent. Although Mini SNMP standard version supports a limited number of MIBs, new MIBs can be added through hard-coded changes. In the current implementation, the following MIBs are used: HOST-RESOURCES-MIB [19], UCD-SNMP-MIB [20] e IF-MIB [21].

Besides information retrieved by the Mini SNMP Daemon agent implementation, information about solar power system and radio parameters are also required. Therefore, the Mini SNMP Daemon implementation was modified so those information could be retrieved using the SNMP protocol as well. Considering radio information, MeshAdmin retrieves transmission power, signal level and transmission rate for each mesh node neighbor. Considering the solar power system, MeshAdmin retrieves solar panel voltage, current generated by the solar panel, batteries voltage, current received by the router, temperature inside the equipment, outdoor temperature and luminosity.

2) Link Data Collection: The Link Data Collection module is responsible for obtaining WMN link information, i.e., source and destination address and link quality value given by the routing metric. That information is retrieved periodically by MeshAdmin.

Link information is given by OLSR-ML routing protocol [15], an OLSR customization [5] that uses ML (Minimum Loss) routing metric in order to obtain the best end-to-end path. OLSR is a proactive protocol used in wireless ad hoc networks, which is based on the link state algorithm. In OLSR networks, each node maintains a complete view of the network topology. Therefore, it is possible to retrieve link quality

[1]http://www.net-snmp.org/
[2]http://freshmeat.net/projects/minisnmpd

information from any node in the network that runs OLSR. In order to avoid the increase of monitoring traffic, MeshAdmin gets that information from the mesh gateway.

Collected information is sent to the Storage Module, for further analysis, and to the Topology Display Module, for real-time topology visualization. The network administrator can adjust data collection round intervals for both modules, using the Admin Panel, depending on his needs.

C. Alert Messages

The Alert module is responsible for notifying the network administrator about network problems during collection rounds. Each event triggers the Alert Module, which generates a message to be displayed by the Display Module.

Those messages are divided in three levels: Critical, Warning and Information. Critical level is responsible for identifying failures in nodes and links, e.g. a disconnected node or a broken link. Warning level alerts are mainly about mistakes in MeshAdmin configuration, e.g. nodes present in the gateway route table that were not added in the configuration interface. Information level alerts provide messages about execution of platform routines, e.g. date and time of information collection or appearance of new links. Since mesh network topology is dynamic, it is interesting to inform the network administrator about the appearance of new wireless links.

D. Storage Module

The Storage module is responsible for storing data retrieved by the Collection Module, data inserted by the network administrator in the Admin Panel and messages generated by the Alert Module.

The Storage Module uses the PostgreSQL[3] relational database, designed to keep relevant information collected by MeshAdmin platform and to quickly answer queries made by the Display Module. MeshAdmin database organization is shown in Figure 2.

MeshAdmin database stores information about mesh networks and nodes that will be monitored. For each monitored mesh node or gateway, an identification (id), its geographical coordinates, its WLAN IP address and WAN IP address in case of gateways are stored. Each node belongs to a mesh network that has a name, a description and geographical coordinates.

Mesh nodes have a set of resources that will be monitored by MeshAdmin. The information retrieved about those resources are divided in (i) Node Measures: it keeps node uptime, available and total memory and CPU Load; (ii) Interface Measures: for each interface, it keeps bytes, packets, discard and errors in and out; (iii) Disk Measures: for each disk, it keeps total, available and used disk space; (iv) Power (Energy) Measures; (v) Radio Measures. These two last measurement groups keep all the information retrieved by the Node Collection Module related to power and radio resources.

Moreover, network links and their quality information are stored. Each link is composed by a source and a destination

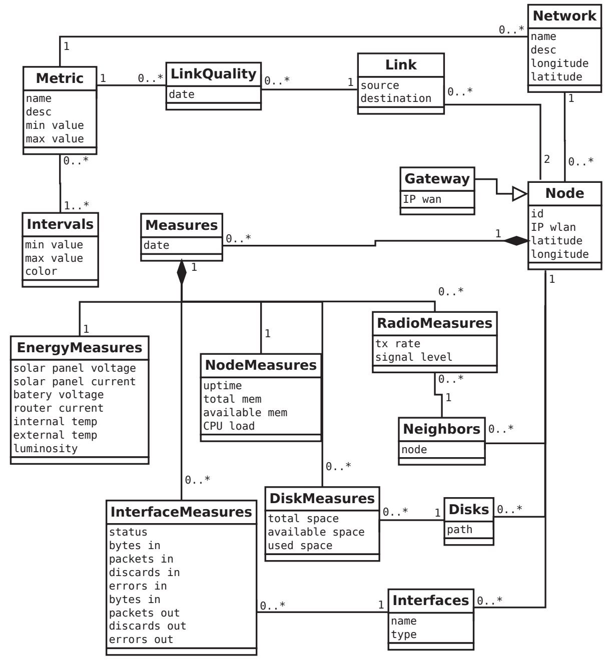

Fig. 2. MeshAdmin database organization.

node and its quality-aware metric. Each metric has a set of numeric intervals that are associated to colors in order to distinguish link quality in the topology visualization.

E. Display Module

In order to facilitate remote monitoring, MeshAdmin offers a web interface. This interface was developed using Django[4], a high-level Python Web framework.

MeshAdmin initial screen has three sections: topology view, node monitoring and alert messages. Besides, this initial screen has a link to a configuration interface. Both initial screen and configuration interface can only be accessed by administrators after authentication. Figure 3 shows a snapshot of MeshAdmin initial screen.

1) Topology Visualization: One of the most important requirements for a WNM management platform is to provide real-time topology visualization. The Topology Display Module shows real-time topology, with network nodes geographically positioned in a background map and network links colored according to its quality (as shown in Figure 3).

Link quality information is retrieved by the Collection Module and processed by a Python script and presented using Google Maps API[5]. Nodes are placed in the map according to their latitude and longitude and links are colored according to a color scale that associates different colors to routing metric value intervals and can be defined by the network administrator.

2) Node Monitoring: In order to facilitate navigation through networks and nodes, on the right side of MeshAdmin

[3]http://www.postgresql.org

[4]http://www.djangoproject.com

[5]http://code.google.com/intl/pt-BR/apis/maps/documentation/javascript/

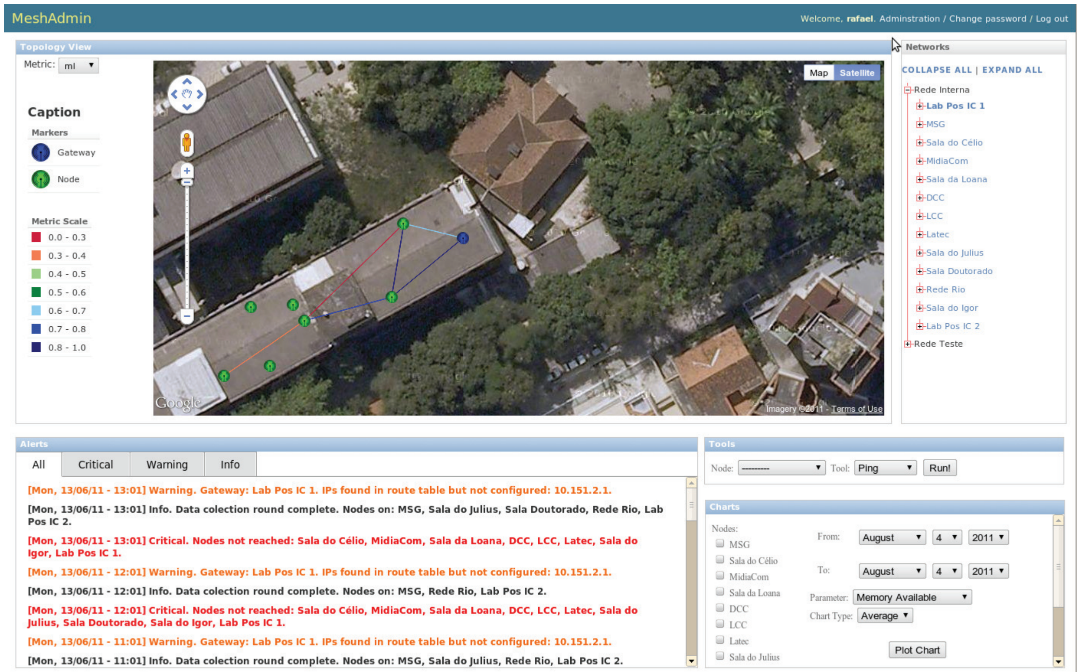

Fig. 3. MeshAdmin initial screen.

main screen (see Figure 3), there is a tree with all networks and nodes monitored by the platform. The network administrator can centralize a node in the topology view by clicking in its tree entry. He can also access latitude, longitude and WLAN IP address by opening a node branch. Moreover, there is a link to access SNMP collected information in each node branch.

Collected information is divided basically in 3 groups: Node Resources, Interfaces and Disks. Node Resource data are uptime, total memory, available memory and CPU load. For each configured interface, the following parameters are collected: bytes in and out, packets transmitted, discarded and lost, in and out. For each disk set, collected information is total, available and used space. If the node is powered by solar panel and batteries - and the network administrator configured the node as solar powered in the Admin Panel -, the Power System group is also displayed showing the information relating to the power system. Figure 4 shows the node information screen of a chosen node in the MeshAdmin platform.

MeshAdmin platform also provides the visualization of data collected from nodes through charts. It is possible to generate charts from the following parameters: LAN and WLAN bytes in and out, available and used memory, CPU load, battery and solar panel voltage, current generated by the solar panel and current received by the router, internal and external temperature and luminosity.

In the lower-right corner (Figure 3), MeshAdmin provides an interface for chart creation, where the network administrator inputs initial and final date of observation period, the parameter to be observed, the nodes that will be observed and the chart type. It is possible to generate bar charts or line charts.

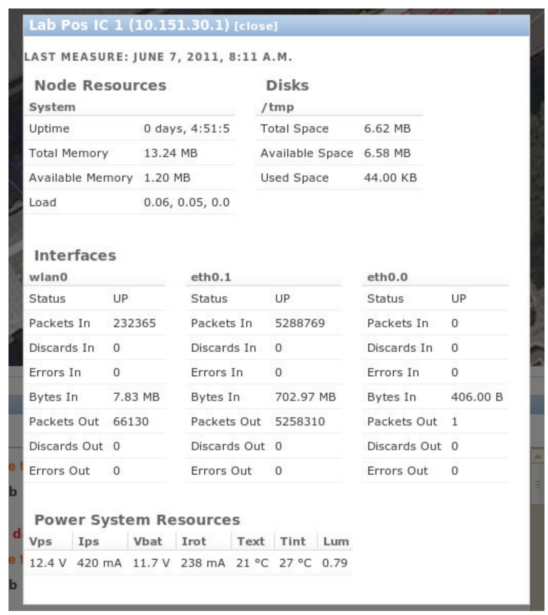

Fig. 4. MeshAdmin node information frame.

Moreover, MeshAdmin provides a measurement tool – placed above the chart frame – that provides to the network administrators the facility of measuring network performance using Ping and Iperf tools through the web interface. With this tool, it is possible to measure throughput, latency and packet loss from the management server to each monitored mesh node.

F. Alert Message Visualization

In the lower-left corner (Figure 3), MeshAdmin presents alert messages generated by collection routines. Those messages are presented in four tabs. The *All* tab presents all messages generated by MeshAdmin. The other tabs, *Critical*, *Warning* and *Info*, present messages corresponding to its level.

V. MeshAdmin Evaluation

The Node Data Collection Module is responsible for collecting information for all nodes in the network. This procedure causes the SNMP agent and manager to insert traffic in the network in order to gather relevant data about each node. In order to evaluate MeshAdmin monitoring traffic overhead, tests considering different scenarios in a real testbed were performed. Five different scenarios with 5, 7, 9, 10 and 12 nodes were considered, as shown in Figure 5. For each scenario, thirty collection rounds with a twenty-minute interval between them were performed. All collection rounds were successfully performed by the platform.

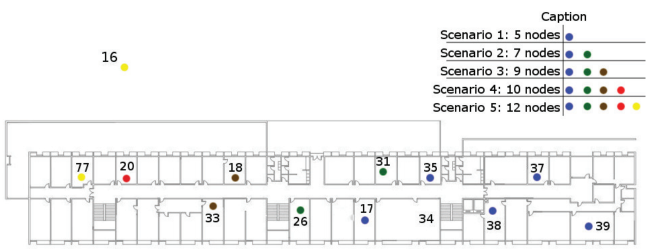

Fig. 5. Evaluation scenarios.

Figure 6 shows the average number of bytes of monitoring traffic inserted by MeshAdmin platform in each scenario. It is also presents an error bar with 95% of confidence interval. As expected, the amount of injected bytes grows linearly as the number of network nodes increases. This happens because each node is queried for the same sort of information individually. In the scenario with 12 nodes, the average overhead traffic was about 24 Kbytes. Considering Wi-Fi channel transmission capacity, its a low amount of traffic and should not interfere on regular data traffic. Analyzing the results, we can infer that the amount of bytes per node is about 2 Kbytes. This means that for a network with 50 nodes, monitoring traffic overhead would be approximately 100 Kbytes.

Figure 7 shows the average duration of collection rounds. It also presents an error bar with 95% of confidence interval. The duration of collection rounds also increases with the number of mesh nodes. However, its growth is not only a function of node quantity. The number of wireless hops from each mesh node to the gateway also interferes in the collection round duration. In our testbed, the network gateway was placed in one end of the network topology (node 39 in Figure 5). Therefore, the average latency between nodes and the network gateway increased when new nodes were added. However, in the 12-node scenario, two nodes were 7 hops away from the gateway, and the average duration of the collection routine was about

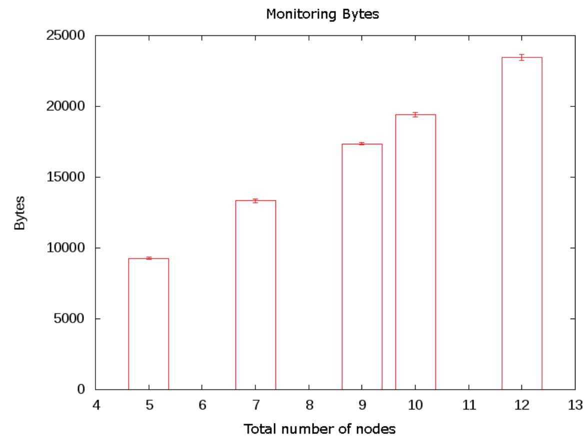

Fig. 6. Monitoring traffic overhead in bytes.

4 seconds. This means that MeshAdmin can collect data from network nodes in each 5 seconds for that scenario, in case there is a need for high granularity information.

Notice that the Link Data Collection Module does not insert any additional traffic in the mesh network, as link information is retrieved from the gateway using its wired interface.

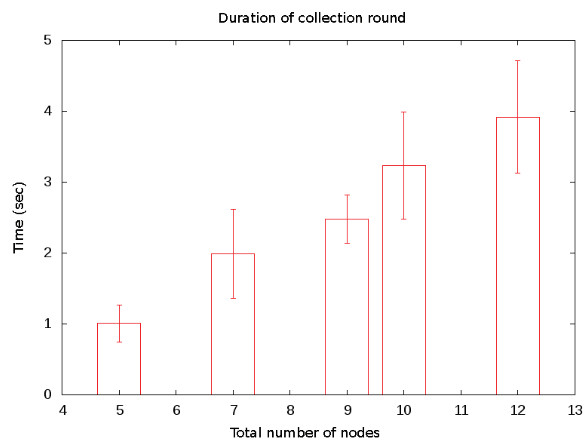

Fig. 7. Collection round duration.

VI. Comparison with Related Work

Sailham et al. [3], Nanda and Kotz [4], Huang et al. [6] and WiMFlow [8] proposed techniques for monitoring different network elements. However, their tools do not provide real-time topology visualization and navigation through network topology.

Jardosh et al. [9] presented SCUBA, which focused on real-time topology visualization and collects several network parameters. However, it does not collect information about network elements, which is useful for the investigation of network failures. Moreover, its topology visualization does not provide geographical position of nodes that is useful for the network administrator to obtain a specific node location.

TABLE I
MESHADMIN COMPARISON WITH RELATED WORK.

	Data Collection			Topology	Fault monitoring		Node
	Users	Nodes	Net	Visualization	Detection	Notification	Config.
MeshAdmin		x	x	x	x	x	
[Sailham et al.]			x		x	x	
MeshMon			x		x	x	
Mesh-Mon	x		x				
MeshFlow	x		x		x		
WiMFlow	x		x		x		
Abaré		x	x		x	x	x
[Jan et al.]		x	x	x	x	x	
Maya							x
SCUBA			x	x	x		
MTV			x	x	x		
WiFiDog	x						
CoovaChilli	x						

MeshAdmin, Abaré [11] and Jan et al. [12] are concerned about node monitoring. While the other presented tools only collect information about network parameters and user flows, MeshAdmin also monitors network nodes. Since nodes have limited storage, memory and CPU processing resources, this is an important feature for monitoring networks with low-cost equipments.

Jan et al. [12] implementations presented many of the desired aspects wanted by MeshAdmin. However, its insertion of monitoring traffic is made each one second by each mesh node. A considerable number of mesh nodes being monitored can harm data traffic, as wireless links have limited capacity. The authors should be concerned about that. Besides, its tool is developed for a custom implementation of a specific wireless network card driver, which restricts its use.

MeshAdmin current implementation already fulfills several previously described requirements as described in Section IV. It already provides network monitoring, with topology real-time visualization and node monitoring, using SNMP MIBs. It also detects and informs the network administrator about network faults through its web interface.

Table I compares MeshAdmin with related work considering the aspects discussed in Section III. The item "Data Collection" considers the capacity of gathering and presenting information about users, nodes and the network itself. The comparison also considers the "Topology Visualization" to be presented in real-time. The item "Fault monitoring" indicates which of the presented tools can detect and notify the administrator about failures in the network. Finally, the item "Node Config." informs which tools are able to automatically configure mesh nodes.

VII. FINAL REMARKS

An integrated management platform allows the network administrator to identify and diagnose network failures without recalling to other management tools. This paper presented the functional requirements of an integrated platform for wireless mesh network management.

This work also described the MeshAdmin platform. Its current implementation satisfies a group of the requirements presented, providing node and network information, topology visualization, fault detection and notification. In addition, MeshAdmin focuses on low-cost mesh access points, therefore node resource constraints were considered in its design. MeshAdmin is already in operation in a mesh network testbed maintained by MídiaCom Lab, at the Fluminense Federal University in the city of Niterói, Brazil.

Moreover, this paper evaluated MeshAdmin in terms of monitoring traffic overhead and showed that overhead traffic is very small and will not interfere with regular data transmission.

MeshAdmin is built using a modular architecture in order to facilitate the deployment of new features. New modules can easily be developed as Django apps and incorporated into MeshAdmin's code. MeshAdmin's database can also be easily extended with new tables if additional information needs to be stored and managed.

As future work, a user control module that includes user flow monitoring and controlling is being developed. This module will collect flow information and display this information graphically through the Display Module.

It is also important to extend the set of alert messages. New messages can alert the network administrator about resources over usage or a significant traffic increase in a specific wireless link. An intelligent mechanism will be deployed to analyze MeshAdmin's database and compare historical information with data collected in real time in order to detect anomalies in networks or nodes.

Finally, as another future work, MeshAdmin will provide a configuration module where some parameters of all mesh nodes or a group of them will be configured automatically, instead of accessing each node that needs configuration.

ACKNOWLEDGMENT

The authors would like to thank FAPERJ, CNPq, CAPES, INCT-MACC and TBE (*Transmissores Brasileiros de Energia*), for making this work possible through research support.

REFERENCES

[1] M. Campista, P. Esposito, I. Moraes, L. Costa, O. Duarte, D. Passos, C. de Albuquerque, D. Muchaluat-Saade, and M. Rubinstein, “Routing metrics and protocols for wireless mesh networks,” *Network, IEEE*, vol. 22, no. 1, pp. 6–12, Jan.-Feb. 2008.

[2] J. Duarte, D. Passos, R. Valle, E. Oliveira, D. Muchaluat-Saade, and C. Albuquerque, “Management issues on wireless mesh networks,” in *5th Latin American Network Operations and Management Symposium (LANOMS 2007)*, Petrópolis, RJ, Brazil, 2007.

[3] F. Sailhan, L. Fallon, K. Quinn, P. Farrell, S. Collins, D. Parker, S. Ghamri-Doudane, and Y. Huang, “Wireless mesh network monitoring: Design, implementation and experiments,” in *IEEE DANMS workshop*, Washington, DC, USA, 2007.

[4] S. Nanda and D. Kotz, “Mesh-Mon: A multi-radio mesh monitoring and management system,” *Computer Communications*, vol. 31, no. 8, pp. 1588–1601, 2008.

[5] T. Clausen, P. Jacquet, A. Laouiti, and et al., “Optimized link state routing protocol (olsr),” *IETF RFC 3626*, October 2003.

[6] F. Huang, Y. Yang, and L. He, “A Flow-Based Network Monitoring Framework for Wireless Mesh Networks,” *IEEE Wireless Communications*, vol. 14, no. 5, 2007.

[7] Netflow, “Cisco IOS Netflow,” 2011, www.cisco.com/web/go/netflow. Accessed in January, 2011.

[8] C. Popi and O. Festor, “Wimflow: a distributed, self-adaptive architecture for flow monitoring in wireless mesh networks,” in *Network Operations and Management Symposium (NOMS), 2010 IEEE*, Osaka, Japan, 2010.

[9] A. Jardosh, P. Suwannatat, T. Hollerer, E. Belding, and K. Almeroth, “SCUBA: Focus and Context for Real-Time Mesh Network Health Diagnosis,” *Lecture Notes in Computer Science*, vol. 4979, p. 162, 2008.

[10] D. Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *ACM MobiCom*, San Diego, CA, USA, September 2003.

[11] B. Pinheiro, V. Nascimento, W. Moreira, and A. Abelém, “Abaré: A deployment and management framework for wireless mesh network,” in *Communications, 2009. LATINCOM’09. IEEE Latin-American Conference on*. IEEE, 2009, pp. 1–6.

[12] F. Jan, B. Mathieu, and D.-E. Meddour, “A monitoring tool for wireless multi-hop mesh networks,” in *Network Operations and Management Symposium (NOMS), 2008 IEEE*, Salvador, Brasil, 2008.

[13] D. Manzano, J. Cano, C. Calafate, and P. Manzoni, “MAYA: A Tool For Wireless Mesh Networks Management,” in *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS 2007)*. IEEE, 2007, pp. 1–6.

[14] R. Valle, D. Passos, C. Albuquerque, and D. Muchaluat-Saade, “Mesh Topology Viewer (MTV): an SVG-based interactive mesh network topology visualization tool,” in *IEEE Symposium on Computers and Communications, 2008. ISCC 2008*, Marrakesh, Marrocos, 2008, pp. 292–297.

[15] D. Passos, D. Teixeira, D. Muchaluat-Saade, L. Magalhães, and C. Albuquerque, “Mesh network performance measurements,” in *International Information and Telecommunications Technologies Symposium (I2TS)*, Cuiabá, MT, Brazil, 2006.

[16] M. Lenczner, “Wireless portals with wifidog,” *Linux J.*, vol. 2005, no. 140, p. 8, 2005.

[17] CoovaChilli, “Open source captive portal access controller and radius software,” 2011, http://coova.org/wiki/index.php/CoovaChilli. Accessed in January, 2011.

[18] C. da Silva, D. Passos, J. Duarte, I. Moraes, and C. V. N. de Albuquerque, “DynTun: A Tool for Providing Multihoming Support in Wireless Mesh Networks ,” in *9th International Information and Telecommunication Technologies Symposium (I2TS’2010)*, Rio de Janeiro, RJ, Brazil, 2010.

[19] S. Waldbusser, “Host resources mib,” *IETF RFC 2790*, March 2000.

[20] University of California Davis, “Ucd-snmp Project Home Page.” 2012, http://http://www.net-snmp.org/docs/mibs/ucdavis.html. Accessed in January, 2012.

[21] K. McCloghrie, “The interfaces group mib,” *IETF RFC 2863*, June 2000.