Network Topology Visualization and Monitoring for Multi–Hop Wireless Networks

Roberto Riggio[1], Matteo Gerola[1], Antonio Francescon[1], Andrea Zanardi[1], Tinku Rasheed[1], and François Jan[2]

[1] CREATE-NET, Via Alla Cascata 56/D, 38123, Povo, Trento, Italy,
name.surname@create-net.org,
[2] France Telecom, R&D Division, 22300, Lannion, France
francois2.jan@orange-ftgroup.com

Abstract. Multi-hop wireless systems represent a viable means for deploying access networks covering medium-size areas with limited investment, making use of commodity hardware and freely available software suites. At the same time, management of such networks represent an overly complex task, due to the joint effect of the time-varying nature of the radio channel, user mobility and the inherently distributed nature of the system. A number of solutions are currently researched, whereby network management functionalities get embedded within the wireless network itself. A common building block of such approaches is represented by a monitoring framework able to bring the relevant network-level information to the decision points. In this paper, we present a distributed network monitoring toolkit, specifically developed for wireless multi-hop networks. The toolkit allows network administrators to monitor the status of the network as well as to plan and execute active measurement campaigns. Information is stored in a distributed network-wide repository and is accessible through a web interface.

Key words: wireless networks, network management, network monitoring, mesh architecture

1 Introduction

Wireless Mesh Networks (WMNs) provide many advantages over traditional wireless networks, such as robustness, greater coverage, low up–front costs and ease of deployment. Despite this, several critical issues need to be addressed in order to turn WMN into a commodity [1] solution for Wireless Internet Service Providers (WISP) operated deployments. In particular, for some usage scenarios, dedicated network control and management appliances may prove impractical due to either cost and/or architectural reasons. As a result, in the last few years, a tendency emerged to distribute network management functionalities within the network itself [2, 3, 4, 5, 6]. The effective deployment of such solutions requires a scalable signaling channel for gathering network status information and conveying it to the relevant decision points, as well as a controller–less network management paradigm where network control and management functionalities are embedded into the network elements themselves (i.e. the access points or mesh routers).

In this paper, we present *OBELIX*, a distributed network monitoring toolkit specifically tailored for infrastructure multi-hop wireless networks, such as IEEE 802.11-based WMNs. The monitoring toolkit is designed to support domain specific knowledge and incorporates appropriate reasoning logic to detect and diagnose faulty network conditions in addition to being capable of performing necessary root cause analysis and autonomic recovery decisions for network administration purposes or for applications. OBELIX enables network administrators to manage network performance, find and solve network problems, and plan for network growth. Its main features are:

– *Adjustable level of pervasiveness.* OBELIX supports different levels of participation in the monitoring efforts. While in traditional network monitoring solutions such as SNMP [7], the function performed by each device is hard–coded at system deployment time, OBELIX exploits a monitoring overlay where each node's participation in the monitoring efforts (i.e. its role) can be dynamically changed at run–time to adapt to changing conditions (e.g., addition of new nodes in the network, nodes becoming unreachable due to faulty links).
– *Interoperability with legacy solutions.* Legacy network management tools are supported through an SNMP interface. The SNMP protocol is used to interact with already deployed SNMP Agents as well as to convey the gathered network state information to an existing SNMP–compatible network management system.
– *Ubiquitous network management.* OBELIX includes an advanced web-based Management Dashboard that allows network administrators to both monitor and manage the network from anywhere using just a web browser. Such Dashboard provides the network administrator with a synthetic representation of the network status in order to enable quick and efficient troubleshooting of critical situations.

The rest of the paper is structured as follows. Section 2 provides an overview of related work and of the state-of-the-art in the field. Section 3 describes the design principles and the implementation choices around which OBELIX has been built. The system architecture is introduced in Sec. 4. The implementation details and the outcomes of an experimental campaign, obtained from a small-scale testbed implementation, are reported in Sec. 5. Finally, Sec. 6 concludes the paper discussing a number of open issues and future extensions.

2 Related Work

A large set of protocols exists to support network and network devices management. Common solutions include SNMP [7], ICMP [8], netconf [9], and capwap [10]. However, most of such tools are designed around centralized architectures meaning that each node participating to the network runs a process which gathers information about the current network state. When a problem is recognized, the running process sends alerts to some management entities. Upon receiving these alerts, the management entities are programmed to react by taking some actions (e.g., operator notification, event logging, system reboot/shutdown, etc.). Management entities can also poll end-stations to check the values of certain variables.

In particular, Single Network Management Protocol (SNMP) represents the most widely used protocol for building monitoring applications [11]. Formally, SNMP is an application layer protocol developed in order to standardize the exchange of management information between network devices. From an SNMP perspective, a network is constituted by a set of managed device (devices which are monitored to gather information on their status), agents (software running on managed devices) and a network management system (software running on managers, i.e., nodes managing the network). The network-level parameters and quantities monitored are termed Managed Object (MO). An example of a MO is the radio channel being used on a given wireless interface. Management information is viewed by SNMP as a collection of Managed Objects, organized as a virtual information repository, called Management Information Base (MIB). Each device participating in an SNMP–managed network maintains a MIB, which is accessed using SNMP. Managers can query the MIB asking for information on a given MO. SNMP uses UDP connections for exchanging data among entities in the monitoring system. While in v2 of the standard, an interface was specified for enabling communication among managers, SNMP is inherently centralized in nature. With OBELIX, we aim at creating a distributed and effective architecture which is SNMP-compatible in order to provide backward compatibility with existing network management systems. In practice, the points at which we will aggregate network information will provide an SNMP–compatible interface, while the processes for gathering and replicating information will not rely on the methods specified in SNMP.

A Distributed Architecture for Monitoring Mobile Networks (DAMON) is introduced in [12]. DAMON relies on agents within the network to actively monitor network behaviour and to send this information to data repositories. DAMON's generic architecture supports the monitoring of any protocol, device, or network parameter. VISUM [13] is a distributed framework for monitoring wireless networks. Data, collected by agents distributed over several host, is gathered at a centralized repository and can be exploited by a visualization tool. In [14], the authors propose a novel monitoring framework capable of dynamically tuning the granularity of the data collection procedures according to some observed events (e.g. threshold crossing). Albeit showing adaptive characteristics from a data gathering perspective, the proposed systems still relies on centralized storage and processing of information. In contrast to the aforementioned works, our approach relies on a fully distributed repository to maintain the global network state and to make it available to all the nodes running network management tasks.

3 System Design

In this section, the three pillars *Adjustable level of pervasiveness*, *Interoperability with legacy solutions*, and *Ubiquitous network management*, upon which OBELIX has been designed are discussed and the implementation choices are introduced.

3.1 Adjustable level of pervasiveness

Typical network management solutions, such as SNMP, rely on a centralized architecture, whereby network management tasks are carried out on a per-device basis by the network administrator. Such a solution results in poor spatial reuse of the wireless medium, congestion of routes to the network controller, and excessive load at the repository itself. OBELIX moves away from this highly centralized approach to network management by embedding management functionality (data analysis and aggregation) into the network itself and by distributing the network state information among the nodes participating in the monitoring efforts improving the availability of the managed information without disrupting network services.

OBELIX supports different levels of participation in the monitoring efforts by the nodes in the WMN. Such participation can be conceptually classified into two categories:

– *Information gathering.* Monitoring agents (Taps in the following) running within a mesh router gather the local network state either by sniffing the traffic flow in their neighborhood (passive approach) as well as by performing on-demand/periodic measurements (active approach).
– *Information analysis.* The local network state information gathered by the Taps is periodically sent to a set of management daemons (Sinks in the following) and exploited to maintain a global view of the network.

It is worth pointing out that information gathering and information analysis are to be considered as two separate, yet non-mutually exclusive functionalities. As a matter of fact, any node in the WMN can support a single functionality, both, or none at all. In the latter case, information about the state of a network devices can only by inferred through passive sniffing from neighboring Taps. Network state updates are delivered by the Taps to the Sinks in the form {`key, value`} pairs, where:

– the `key` field uniquely identifies the managed object;
– the `value` field holds the actual value of the managed object;

Managed Objects are defined as Global or Local through a configuration file. The former identifies pieces of information that have a network-wide scope, i.e. the geographical position of the network devices, while the latter is used to identify pieces of information that are cluster or node specific, i.e. the number of bytes transmitted over an interface.

3.2 Interoperability with legacy solutions

Interoperability with other network management tools will be provided by means of an SNMP-compatible interface. The SNMP protocol will be used to convey management information between Sinks and existing network management systems. An additional HTTP interface will be developed in order to allow the web-based management dashboard to interact with the network Sinks. This section will detail the information exported by network Taps and network Sinks.

3.3 Ubiquitous network management

The Web-based OBELIX Management Dashboard allows network administrators to both monitor and manage the network from anywhere using just a web browser. Such Dashboard supports a combined reactive/proactive approach to network monitoring:

– *Reactive.* Aims at detecting a fault only after it occurs by passively studying its effects on the network. In this case, the Tap will send a failure event (e.g. a node leaves the network or is out of order) to its Sink. Such an event will be logged and made available to the network administrator through the Dashboard.
– *Proactive.* Exploits temporal trends of the monitored properties in order to foresee potential failures and isolate fluctuating behaviors. In this case, the Sink receives regular network state updates from its Tap and from neighboring Sink (e.g. high link utilization or low battery level) and take the appropriate actions (e.g. report a possible congestion situation).

4 System Architecture

The building blocks of the OBELIX toolkit and their relationships are sketched in Fig. 1. It consists of three main blocks, the Tap, the Sink, and the Management Dashboard; their detailed description follows. It is worth noticing that, unlike traditional SNMP–based systems, where the NMS continuously polls every managed device, in our architecture, network state updates are transmitted by the Taps to the Sinks periodically and only if required (i.e. if there has been a change in the state of a managed object).

4.1 Tap

The Tap is a software process running in each managed device. A Tap has local knowledge of the network. Such a knowledge is collected and published an the asymmetric group communication system which implements the OBELIX Signaling layer. The local network state is collected using a modular based information gathering back-end. Domain-specific information (e.g. routing tables, link status, etc.) are collected by specialized plugins and presented to the Tap using a protocol agnostic representation. The following plugins are currently supported:

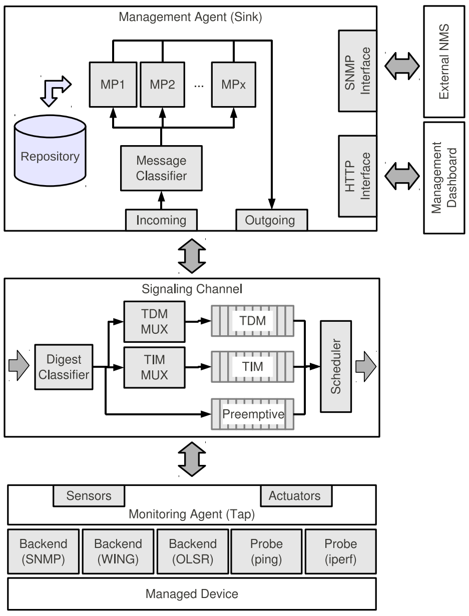


Fig. 1: OBELIX architectural components: Tap, Sink, Signalling Layer, Dashboard, and their interactions.

– Routing WING plugin. Use to provided support for the WING routing protocol. WING [15] is a DSR-like routing protocol derived from the Roofnet [16] project by the MIT and optimized for network scalability and throughput rather than for supporting mobility. Wing extends the original protocol by adding support for multiple radio interfaces and for link quality routing using either the ETX, the ETT, ot the WCETT metric [17].
– Routing OLSR plugin. The Optimized Link State Routing Protocol (OLSR) is a level-3 routing protocol optimized for mobile ad-hoc networks, but can also be used on other wireless ad hoc networks.
– Probe Ping/Iperf plugins. Allows network administrators to plan, execute and retrieve network measurements campaigns using the Ping/IPerf utility. Measurements campaign can be planned for execution at a certain time. Results of the campaign are available trough the web interface after the campaign has terminated.

4.2 Sink

The Sink is a software process running on a subset of the nodes composing the WMN. A Sink has global knowledge of the network state, which is stored on a shared repository. Such a repository is made accessible to the Management Dashboard through an HTTP interface. At bootstrap, Taps selects one Master

Sink for their normal operations and zero, one or more Slave Sinks to be used if the Master Sink fails.

The repository is implemented in the form of a distributed database and provides a WMN-wide knowledge base. The data model for the shared repository is reported in Fig. 2 as UML Class Diagram describing the data model. The SQLite database has been chosen to implement the OBELIX data model. The data model can be easily extended in order to gather additional information. As a matter of fact, the actual data model currently used by the OBELIX Monitoring Toolkit is entirely defined in a set of configuration files. New Managed Objects can be defined by the network administrator. In such a case, an helper script which effectively gathers the information from the managed device must be provided.

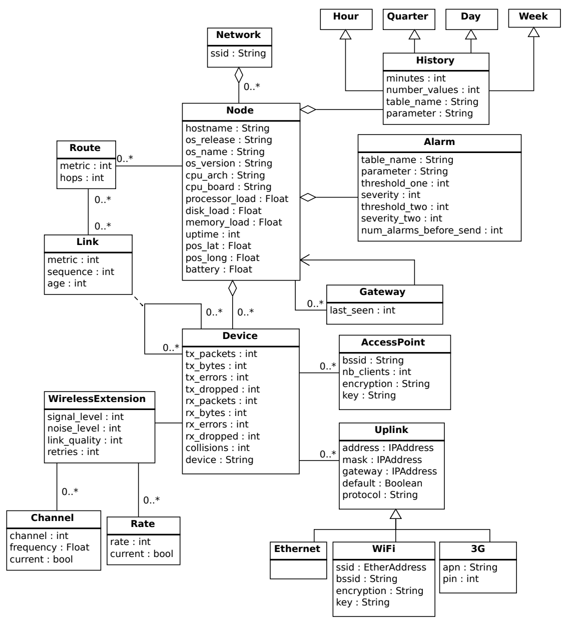


Fig. 2: OBELIX Data model.

It is worth noticing that user-defined extensions are not mandatory and can thus be deployed only on a sub-set of the nodes participating the WMN. On the other hand, all nodes in an OBELIX-managed network are required to implement at least the data model reported in Fig. 2.

4.3 Signaling layer

The OBELIX signaling layer is implemented in the form of a scalable asymmetric publish/subscribe system where messages are differentiated according to their temporal properties. For example, consider a requirement such as tracking the network topology in real-time. If topology information from the network is not delivered for processing in a timely manner, the resulting view of the network can be inaccurate. On the other hand, another requirement could be to monitor the amount of traffic forwarded by a node without any constraints on time. Based on their temporal properties, network state updates generated by the Taps can be classified into three categories:

– *Preemptive Message.* Alarms generated by the monitoring daemons that require immediate forwarding. Examples of Preemptive Messages include notification of a node going off–line, communication links becoming congested or fluctuating.
– *Time Dependent Message (TDM).* Delay sensitive message that can tolerate a fixed transmission delay. Examples of TDM include link state updates, number of users currently associated to a CPE etc. Since such network state updates are typically small in size, they can be aggregated at intermediate hops in order to improve the wireless medium utilization.
– *Time Independent Message (TIM).* Delivered on a best effort fashion only where and when the network bandwidth is not required by other applications. These kind of messages are expected to be larger in size than TDMs (transmission logs, extended link utilization statistics, etc.). With respect to this kind of messages, the monitoring tool is expected to support proper data fragmentation procedures in order to handle TIMs that are larger than the networks maximum transmit unit (MTU). A possible choice is to take direct advantage of the packet fragmentation capabilities already provided by the IP protocol.

In order to control the amount of signaling traffic exchanged over the signaling, OBELIX supports several levels pervasiveness:

– *Non-pervasive.* Collection of the network state is done by a single network element, which implements both Information Gathering and Information Analysis functionalities, while the other nodes in the network implement only Information Gathering functionalities. Such an approach is conceptually equivalent the an SNMP-based network management architecture where a single network element periodically interrogates network devices and revives alarms and events. In this setup administrative tasks are simplified in that all the information are already available at a single point. On the other hand, having a single global repository results in a poor spatial reuse of the wireless medium and introduces a single point of failure in the system.
– *Fully pervasive.* Collection and analysis of the network state is done by every node in the network. Such an approach effectively creates a distributed repository holding the global network state information where every node has a global knowledge of the network. While delivering the highest level of resilience to network failures, this setup is characterized by an high signaling overhead in that network state updates must be circulated among all the node participating in the monitoring efforts.

– *Hybrid.* The degree of pervasiveness of the previous solution can be minimized by limiting the nodes implementing Information Analysis functionalities, i.e. the Sinks, and by grouping the nodes implementing Information Gathering functionalities, i.e. the Taps, into clusters. In this configuration, each cluster is composed by a variable number of Taps and one Sink, which acts as cluster head.

In the latter scenario, the one supported by default by OBELIX, Sinks have a complete knowledge about the state of the nodes in their cluster, while information about other clusters is limited to Global Managed Objects. Such an architecture allows for several degrees of pervasiveness by simply changing the ratio between the Taps and the Sinks and by tagging Managed Objects as either Global or Local. A typical configuration for a WMN would involve Taps deployed in each node and Sinks deployed only on the network gateways. Likewise, only information such as the geographical position of the devices and their default route to the mesh gateways would be tagged as Global. This is due to the fact that most of the traffic in a WMN flows through a set of pre-determined, non-mobile and fully-capable nodes (the mesh gateways).

4.4 Management Dashboard

The global network state information made available by the Sinks is exploited by the Management Dashboard in order to provide the network administrator with a powerful web-based network management interface. In particular each Sink implements an HTTP interface which can be exploited to send queries and commands to gather the value of a Managed Object or to request the execution of configuration and monitoring actions.

The dashboard is implemented in the form of a web application exploiting AJAX as enabling technology. The rationale behind this choice is to move as much of the computation on the client side as possible, while still using Web-based technologies allowing the access from any host. In fact, common mesh routers are characterized by relatively limited computing power in terms of both processor speed and RAM. On the other hand, clients are typically characterized by powerful processors equipped with adequate memory.

Due to these architectural choices, it is possible to deploy the Management Dashboard on any mesh router in the network as well as on an external machine. In the latter case, the machine hosting the Management Dashboard must have access to the HTTP interface exported by one of the Sinks. Security is handled using the HTTP authentication facilities provided by the web server.

The Dashboard provides the network administrator with a synthetic representation of the network status in order to enable quick and efficient troubleshooting of critical situations. The Dashboard supports the following features:

– *Display topological and geographical information.* A Google Maps-based interface is exploited. If no geographical information is available, a simple graphical representation of the network topology (with randomly placed nodes) is provided. Network administrator(s) can navigate the graphical representation of

the network in order to identify critical spots and to gather detailed information about the managed devices.
- *Provide detailed reports of the network's performance.* The dashboard allow the network administrator(s) to sort the collected data by protocol, device, and other factors. Performance trends and network resources utilization pattern trends can be used to identify nodes with intermittent connections that need attention.
- *Raise accurate alarms.* Network administrator(s) are notified when there is a significant network event, such as a node going off-line, a gateway connection failing, etc. These functionalities are implemented within the Sink and are exported using either emails or an RSS feed.
- *Support network profiling.* Network administrator(s) are empowered with a set of tools to run real-time bandwidth and latency test. In this scenario, the Dashboard is expected to act as a remote front-end to traditional network probes such as: iperf, ping, etc. A detailed report of the measurement campaign is made available through the Dashboard after the tests have been completed. Traffic traces pre-processing is performed on the server-side (i.e. where they are collected) in order to lower the network usage.
- *Implement administrative tasks.* The Dashboard allows network administrator(s) to implement administrative tasks. For example, the Dashboard allows to modify the value of configuration parameters, such as the frequency of an hotspot, or to change the role of a node in the monitoring overlay as well as to deploy firmware updates to all the nodes automatically.

5 Evaluation

In this section, we report the outcomes of some experimental tests conducted using a prototypical implementation of OBELIX over a small-scale (15 nodes) wireless mesh network testbed. Our goals for this OBELIX-monitored mesh deployment were twofold. On the one hand we wanted to validate the design and implementation of the toolkit in a realistic environment by tracking the topology of the network in real-time. On the other hand we wanted to perform a set of measurements campaigns using the probing facilities provided by the toolkit itself.

5.1 Implementation Details

The OBELIX distributed monitoring toolkit is implemented in Python, a lightweight interpreted programming language. Signaling traffic is exchanged over an asymmetric scalable group communication system that allows the different entities involved in the monitoring efforts to communicate with one another supporting the entire monitoring life-cycle. Within a single OBELIX–monitored node, up to three distinct software processes can be active at any given time: the *sink*, the *tap*, and the *communication channel.* The latter effectively implements the message scheduling and processing on an hop-by-hop basis.

All nodes in the OBELIX overlay run the *communication channel* process. Nodes that are also publishers and subscribers of monitoring information will also run the *tap* and the *sink* processes, respectively. Albeit not exploited during our measurement campaign, a node can implement only data dissemination functionalities by running just the *communication channel* process. Standard TCP and UDP sockets are used to enable communications among processes. Such a feature may prove useful in a scenario where we are not interested in information coming from every device, yet we want to take advantage of their forwarding capabilities. Internet sockets are used to enable inter–process communications.

Messages being dispatched by the *communication channel* are composed by a header and a body. The header is used to indicate how and where a message should be delivered and the body provides information and commands to the destination entity. The header includes, among the other information, source and destination address, together with an indication of the type of message it carries (necessary for ensuring appropriate processing at the Sinks). Our goals for this Obelix-monitored WMN deployment were twofold. On the one hand we wanted to validate the design and the implementation of the toolkit in a realistic environment by tracking the topology of the network in real-time and by performing a set of measurements campaigns using the probing facilities provided by the toolkit itself.

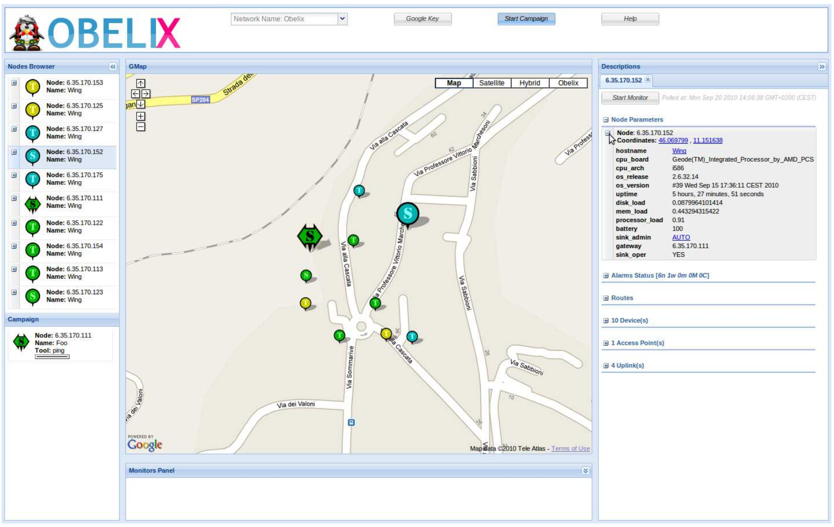
5.2 Experimental Settings and Configuration

The software prototype has been experimentally evaluated over a real–world IEEE 802.11–based mesh testbed built using off–the–shelf components and consisting of 15 multi–radio mesh routers deployed across three floors of a typical office building. Mesh routers are built around three different hardware platforms, namely the PCEngines ALIX 2C2 (500MHz x86 CPU, 256MB of RAM) processor board, the PCEngines WRAP 1E (233MHz x86 CPU, 128MB of RAM) processor board and the Gateworks Cambria GW2358-4 (667MHz ARM CPU, 128MB of RAM). Each node is equipped with two IEEE 802.11a/b/g wireless interfaces (Atheros chipset) with RTC/CTS disabled. Routers employ the OpenWRT operating system, a Linux distribution specifically tailored to embedded devices. Routing is implemented using the Click modular router [18].

5.3 Results

In the scenario accounted in the paper, the OBELIX Dashboard is hosted by the nodes acting as mesh gateways. The network administrator can then connect to any of them from using a regular web browser in order to monitor the status of the network and/or to perform administrative tasks.

Figure 3 shows the Home Page of the Obelix Dashboard. As it can be seen, the interface is partitioned into three panes. The left pane lists the nodes currently available in the network indicating whenever the node is a mesh gateway (double-arrow marker) or a mesh router (round marker). Sink (S) and Tap (T) status is also indicated. The central pane is completely dedicated to the Google Map used to display the real-time network topology. Finally the right pane is used to display contextual information relative to the currently selected node. The interface is highly interactive, as for example, selecting one node on the left pane with a single click will center the map on the node and will display the local information about the same node on the right pane (no additional traffic is generated on the network for this operation). On the other hand, a double click on a node's marker will load both the local and the global information thus generating additional traffic over the network.



Nodes Browser

Node: 6.35.170.153	Name: Wing
Node: 6.35.170.125	Name: Wing
Node: 6.35.170.127	Name: Wing
Node: 6.35.170.152	Name: Wing
Node: 6.35.170.175	Name: Wing
Node: 6.35.170.111	Name: Wing
Node: 6.35.170.122	Name: Wing
Node: 6.35.170.154	Name: Wing
Node: 6.35.170.113	Name: Wing
Node: 6.35.170.123	Name: Wing

Campaign

Node: 6.35.170.111
Name: Foo
Tool: ping

Descriptions

6.35.170.152

Start Monitor Polled at: Mon Sep 20 2010 14:06:38 GMT+0200 (CEST)

Node Parameters

Node: 6.35.170.152
Coordinates: 46.069799 , 11.151638

hostname	Wing
cpu_board	Geode(TM)_Integrated_Processor_by_AMD_PCS
cpu_arch	i586
os_release	2.6.32.14
os_version	#39 Wed Sep 15 17:36:11 CEST 2010
uptime	5 hours, 27 minutes, 51 seconds
disk_load	0.0879964101414
mem_load	0.443294315422
processor_load	0.91
battery	100
sink_admin	AUTO
gateway	6.35.170.111
sink_oper	YES

Alarms Status [6n 1w 0m 0M 0C]

Routes

10 Device(s)

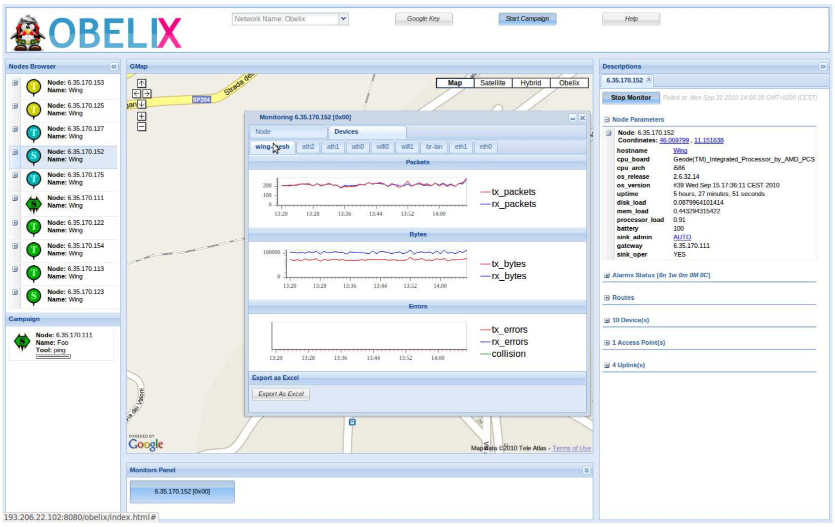
1 Access Point(s)

4 Uplink(s)

Fig. 3: The OBELIX Dashboard: the three panes home page. The dashboard is hosted by the mesh gateways and is accessible using any recent web browser.

The Dashboard can also be used in order to analyse the historical trends for any OBELIX-monitored network object (i.e. transmitted or received packets, signal-to-noise ratio, CPU or memory load, etc) in a graphical form. Figure 4 shows the pop-up through which the network administrator can select the attributes to be visualized. It is worth noticing that these parameter are constantly monitored by each Sink in an OBELIX-monitored network. The sampling period and the number of samples to be stored and user-configurable parameters. For example, in the default configuration the CPU load is sampled every minute and the last 60 samples are stored by each Sink. On top of this statistic, both the hourly and the daily averages are computed and, respectively, the last 24 and 30 samples are stored by each Sink.

OBELIX allows the network administrator to plan and execute network wide measurement campaigns using the embedded network probing facilities. The current version of OBELIX supports two types of network probes: *ping* and *iperf*.



Node: 6.35.170.153 **Name:** Wing
Node: 6.35.170.125 **Name:** Wing
Node: 6.35.170.127 **Name:** Wing
Node: 6.35.170.152 **Name:** Wing
Node: 6.35.170.175 **Name:** Wing
Node: 6.35.170.111 **Name:** Wing
Node: 6.35.170.122 **Name:** Wing
Node: 6.35.170.154 **Name:** Wing
Node: 6.35.170.113 **Name:** Wing
Node: 6.35.170.123 **Name:** Wing

Campaign
Node: 6.35.170.111
Name: Foo
Tool: ping

Polled at: Mon Sep 20 2010 14:06:38 GMT+0200 (CEST)

Node Parameters

Node: 6.35.170.152
Coordinates: 46.069799 , 11.151638

hostname	Wing
cpu_board	Geode(TM)_Integrated_Processor_by_AMD_PCS
cpu_arch	i586
os_release	2.6.32.14
os_version	#39 Wed Sep 15 17:36:11 CEST 2010
uptime	5 hours, 27 minutes, 51 seconds
disk_load	0.0879964101414
mem_load	0.443294315422
processor_load	0.91
battery	100
sink_admin	AUTO
gateway	6.35.170.111
sink_oper	YES

Alarms Status [6n 1w 0m 0M 0C]
Routes
10 Device(s)
1 Access Point(s)
4 Uplink(s)

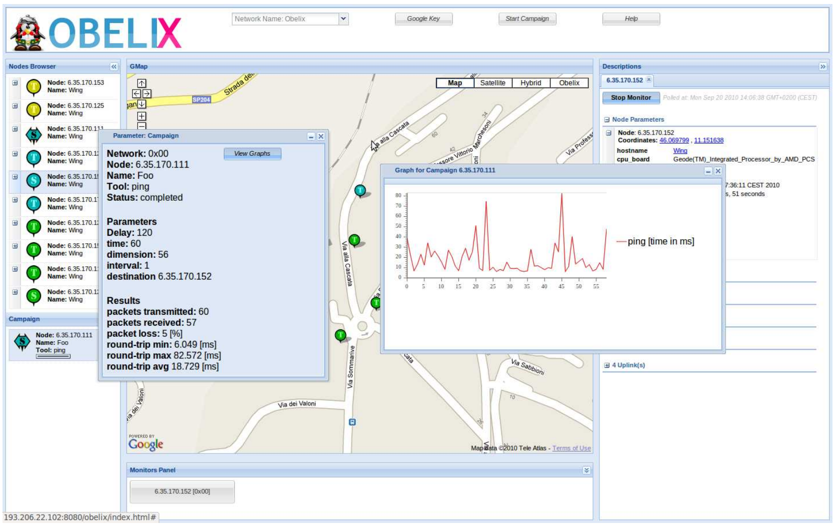
Fig. 4: The OBELIX Dashboard: monitoring historical trends. This figure reports the amount of traffic that traversed the mesh interface in the last hour. The graphs are update in real–time.

The former tool (*ping*) allows the user to performs simple connectivity and round–trip–time measurements campaigns, while the latter tool (*iperf*) allows the user to perform more complex tests involving throughput, jitter and exploiting either UDP or TCP as transport technology,

6 Outlook and Future Work

In this paper, we have introduced a distributed network monitoring toolkit specifically tailored for infrastructure multi-hop wireless networks, such as IEEE 802.11-based Wireless Mesh Networks. Design choices have been made to accommodate the peculiarities of wireless multi-hop networks, in terms of adaptivity, robustness and efficiency requirements. The proposed framework has been prototyped and experimentally evaluated on a 15–nodes wireless mesh network.

As future research directions we plan to exploit concepts and techniques borrowed from the information–centric networking domain in order to make the entire monitoring framework address-agnostic, as well as to use cross–layer techniques in order to handle the replication of global monitoring information across multiple sinks leveraging knowledge on the underlying wireless technology employed.



Node: 6.35.170.153 **Name:** Wing
Node: 6.35.170.125 **Name:** Wing
Node: 6.35.170.111 **Name:** Wing

Node: 6.35.170.111
Name: Foo
Tool: ping

Network: 0x00
Node: 6.35.170.111
Name: Foo
Tool: ping
Status: completed

Parameters
Delay: 120
time: 60
dimension: 56
interval: 1
destination 6.35.170.152

Results
packets transmitted: 60
packets received: 57
packet loss: 5 [%]
round-trip min: 6.049 [ms]
round-trip max 82.572 [ms]
round-trip avg 18.729 [ms]

Node Parameters
Node: 6.35.170.152
Coordinates: 46.069799 , 11.151638
hostname Wing
cpu_board Geode(TM)_Integrated_Processor_by_AMD_PCS

Fig. 5: The OBELIX Dashboard: planning and executing network wide measurement campaigns. This figure shows the results of a campaign exploting the *ping* tool. Results can be exported in CSV format for further processing.

References

1. R. Bruno, M. Conti, and E. Gregori, "Mesh Networks: Commodity Multihop Ad Hoc Networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 123 – 131, Mar. 2005.
2. G. Pavlou, "On the evolution of management approaches, framework and protocols: A historical perspective," *Journal of Network and Systems Management*, vol. 15, pp. 425–445, 2007.
3. R. Mortier and E. Kiciman, "Autonomic network management: Some pragmatic considerations," in *Proc. of ACM SIGCOMMM*, Pisa, Italy, 2006.
4. S. Dobson, S. G. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli, "A survey of autonomic communications," *ACM Trans. Auton. Adapt. Systems*, vol. 1, no. 2, pp. 223–259, 2006.
5. J. N. de Souza and J. Strassner, "Self-organization and self-management in communications as applied to autonomic networks," *Computer Communications*, vol. 31, no. 13, pp. 2935–2936, 2008.
6. A. Gonzalez Prieto, D. Dudkowski, C. Meirosu, C. Mingardi, G. Nunzi, M. Brunner, and R. Stadler, "Decentralized in-network management for the Future Internet," in *Proc. of IEEE ICC – Communications Workshops*, Dresden, Germany, 2009, pp. 1–5.
7. W. Stallings, "Snmp and snmpv2: the infrastructure for network management," *IEEE Communications Magazine*, vol. 36, no. 3, pp. 37 –43, mar. 1998.
8. J. Postel, "Internet control message protocol," IETF RFC 0792, Sep. 1981, http://www.ietf.org/rfc/rfc0792.txt.
9. R. Enns, "Netconf configuration protocol," IETF RFC 4741, Dec. 2006, http://www.ietf.org/rfc/rfc4741.txt.

10. M. Montemurro and D. Stanley, “Control And Provisioning of Wireless Access Points (CAPWAP),” IETF RFC 5415, Mar. 2009, http://www.ietf.org/rfc/rfc5415.txt.
11. J. Case, M. Fedor, M. Schoffstall, and J. Davin, “A simple network management protocol,” IETF RFC 1157, May 1990, http://www.ietf.org/rfc/rfc1157.txt.
12. K. Ramachandran, E. M. Belding-Royer, and K. C. Almeroth, “DAMON: A distributed architecture for monitoring multi-hop mobile networks,” in *Proc. of IEEE SECON*, Santa Clara, California, USA, 2004.
13. C. C. Ho, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, “A scalable framework for wireless network monitoring,” in *Proc. of WMASH*, Philadelphia, Pennsylvania, USA, 2004.
14. R. Raghavendra, P. Acharya, E. Belding, and K. Almeroth, “Antler: A multi-tiered approach to automated wireless network management,” in *IEEE INFOCOM Workshops*, Phoenix, AZ, USA, 2008.
15. F. Granelli, R. Riggio, T. Rasheed, and D. Miorandi, “WING/WORLD: An Open Experimental Toolkit for the Design and Deployment of IEEE 802.11-Based Wireless Mesh Networks Testbeds(,” *Eurasip Journal on Wireless Communications and Networking*, vol. 2010, 2010.
16. J. Bicket, D. Aguayo, S. Biswas, and R. Morris, “Architecture and Evaluation of an Unplanned 802.11b Mesh Network,” in *Proc. of ACM MOBICOM*, Cologne, Germany, 2005.
17. R. Draves, J. Padhye, and B. Zill, “Comparison of Routing Metrics for Static Multi-Hop Wireless Networks,” in *Proc. of ACM SIGCOMM*, Portland, Oregon, USA, 2004.
18. E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, “The Click modular router,” *ACM Transaction on Computer System*, vol. 18, no. 3, pp. 263 – 297, Aug. 2000.