Methodology and Tools for Measurements on Wireless Testbeds: The NITOS Approach

Dimitris Giatsios[1,2], Apostolos Apostolaras[1,2], Thanasis Korakis[1,2], and Leandros Tassiulas[1,2]

[1] CERTH, The Centre for Research & Technology Hellas, 78 Filikis Etaireias Str, 38334, Volos, Greece
[2] University of Thessaly, Department of Computer & Communication Engineering 37 Gklavani - 28th October Str, 38221, Volos Greece
{gidimitr,apaposto,korakis,leandros}@inf.uth.gr
http://nitlab.inf.uth.gr

Abstract. Since its establishment in 2009, the Network Implementation Testbed using Open Source drivers (NITOS) wireless testbed has been extensively used in several research projects for the experimental evaluation of protocols and algorithms. Collection of accurate measurements is of crucial importance for testbed users, allowing them to both select appropriate topologies for their experiments and assess the performance of their implementations. In this chapter, we describe measurement methodologies and tools used in the NITOS testbed in the context of its involvement in some of these projects. We provide examples demonstrating the utility of these tools for specific experiments. We also explain the challenges posed by the complex wireless medium, summarize lessons learned and outline future plans of NITOS towards an enhanced and more integrated measurement framework.

Keywords: Testbed, Experimentation, Measurements, Tools, Management.

1 Introduction

The rapid proliferation of network testbeds around the global research community during recent years is considered as a decisive step for the transition towards the Internet of the future [1]. Experimentation with real network equipment under real conditions addresses the traditional reluctance of the industry to seriously take into account innovative algorithms and architectures tested only in simulation platforms. Programmability of network components and use of open-source technology creates a broad field for hands-on innovation, motivating the interest and involvement of a continuously growing group of researchers. Nowadays, testbeds are not just popular, but thought of as the de facto performance evaluation instruments in network engineering research.

Experience from other research fields, where experimentation is the cornerstone of progress and development, such as Physics or Biology, has shown that

L. Fàbrega et al. (Eds.): FIRE Measurement Workshop 2012, LNCS 7586, pp. 61–80, 2013.

the ability to capture accurate measurements is the most crucial component of the experimentation procedure. Indeed, these fields owe large part of their scientific rigor in the use of sensitive measurement equipment and careful methodology.

Measuring experiment variables and collecting observations are an essential part of any scientifically sound evaluation or comparison of technologies, services, or systems being studied. Furthermore, collected measurements provide testbed operators and future experimenters valuable information on the usage of experimental resources, which may influence long-term management decisions or short-term resource selection for a particular experiment.

In this chapter, we describe the measurement methodologies and specific tools used in the NITOS wireless testbed, one of the key testbeds of the FIRE [2] facility. The wireless environment, due to its broadcast nature and inherent uncertainty, poses particular challenges, affecting the experimental procedure. These challenges, as we will see, call for even more advanced and complicated measurements. Thus, many of the tools we describe in this chapter are particularly focused on wireless testbeds.

The remainder of this chapter is structured as follows. In section 2 we provide a brief presentation of the NITOS testbed. In section 3 we focus on measurement tools used for topology assessment, while in section 4 we describe measurement tools used for the evaluation of experiment results. In section 5, we provide some examples emanating from specific experiments conducted in NITOS in the context of research projects, explaining the respective measurement handling procedures. In section 6, we analyze some challenges related to testbed measurements, based on our experience in wireless testbeds, and present the future plans of NITOS in this field. Finally, we conclude the chapter in section 7, summarizing the key points made.

2 The NITOS Testbed

The history of NITOS stretches back to its establishment in 2009, and deals principally with the need to provision and maintain a remotely accessible infrastructure for experimentation and validation in the emerging field of wireless networks. The testbed was built as an outdoor local network with standalone nodes, running on Linux and featuring Wi-Fi cards supporting open-source drivers. Since then it has grown both in size and diversity of features, most notably with the addition of software-defined-radio hardware and an OpenFlow [3] wired experimental network. It has been publicly available through the Internet almost since the beginning and has hosted several experiments, especially in the area of Wi-Fi. Today NITOS is part of the OneLab European testbed facility [4] and the main wireless testbed of the OpenLab project [5], a key project aiming at the federation and further development of network experimental facilities in Europe.

2.1 Description and Architecture

NITOS [*http://nitlab.inf.uth.gr/NITlab/*] is an integrated testbed with heterogeneous features, that focuses on supporting experimentation-based research in the area of wireless networks. It consists of 40 wireless nodes, situated on the two top floors and the rooftop of a campus building (cf. Fig. 1). The nodes are equipped with commercial Wi-Fi cards that are based on Atheros chipsets and support the MadWiFi/Ath5k/Ath9k open source drivers. Through the open source drivers and the Linux operating system running on the nodes, a researcher can modify the MAC (Media Access Control) and the network layer of the nodes and develop new protocols that are directly compatible and comparable with IEEE 802.11 commercial products. 9 of the nodes are connected to software radio Universal Software Radio Peripheral (USRP) boards (GNU-radio). 15 of them are equipped with USB-cameras and 20 of them with temperature and humidity sensors. 10 nodes are equipped with 802.11n cards that allow for experimentation on a Multiple Input Multiple Output (MIMO) setup. NITOS also features three mobile nodes (two mounted on robots and one moving on rails), in order to test mobility-based scenarios.

The testbed's infrastructure is currently being significantly upgraded. A new set of nodes with advanced hardware features, recently assembled in NITOS, has been added to its equipment. These will form an additional indoor wireless testbed, which will be deployed in a campus facility featuring a fairly isolated environment in terms of wireless interference. A small subset of these new nodes has been developed in a more compact design and is aimed for mounting and operation in vehicles. Moreover, NITOS is being extended to support metro-scale wireless communications. In particular, one Worldwide Interoperability for Microwave Access (WiMAX) base station and two Long Term Evolution (LTE) base stations will be soon deployed and installed. Client devices with WiMAX and/or LTE connectivity will also be added to the testbed's infrastructure. Part of them will be deployed as WiMAX/LTE network interface cards in the existing NITOS nodes. Apart from those, handsets with WiMAX/LTE interfaces will be acquired, to be used by real volunteers, in order to experiment with real mobility scenarios.

The architecture of the NITOS testbed is illustrated in Fig. 2. NITOS is comprised of three discrete wired local networks. The first one is the control network, used for logging into the nodes via the NITOS server, sending and receiving OMF/OML traffic, and for some other simple management tasks. The second one is the Chassis Manager (CM) network. Each standalone node in NITOS features an independently powered microcontroller board called CM card, whose main duty is to power the node on or off. These boards feature an Ethernet interface connected to the CM network, through which remote out-of-band power management of the nodes can take place. The third wired local network is an OpenFlow-capable experimental network. This offers the ability to program customized routing schemes inside this network. This network is virtualized by using the FlowVisor tool [6], so that each user can only direct traffic towards resources belonging to his/her slice. In fact, there is a one-to-one correspondence

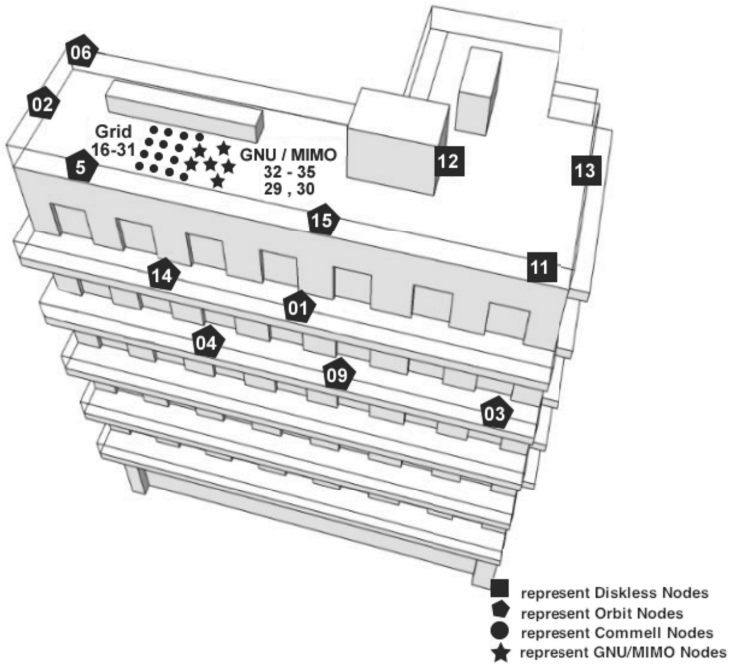


Fig. 1. Blueprint of the NITOS testbed deployment in the ECE Dept. building of University of Thessaly

between OpenFlow switch physical ports and standalone wireless NITOS nodes, therefore there is no need to reserve OpenFlow resources separately.

Apart from those three wired local networks, there is also the wireless network that is formed by the Wi-Fi-enabled nodes and is primarily used by the users for experimentation purposes.

A central server, called NITOS server, is used to host those networks and also serves as the gate where experimenters enter and gain access to the testbed resources. Moreover, NITOS offers access to a second server, called the Development server, that is internally connected to the NITOS server and can be accessed by registered users. The rationale behind the use of the Development Server is to move high performance demand activities, such as driver compilation and development, out of the main NITOS server.

2.2 Testbed Software

NITOS is remotely accessible through a sophisticated managerial system, developed in University of Thessaly (UTH), called NITOS Scheduler [7]. The front-end of this system is a web interface accessible by registered users through the NITOS website, that allows a researcher to reserve some of the testbed's resources for a specific time duration. At the backend, a set of scripts running on the NITOS main server take care of authorization and access control, based on the reservations.

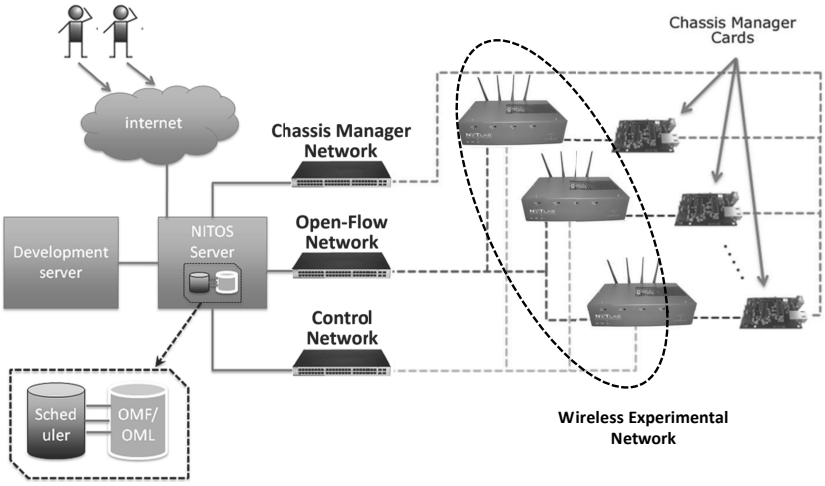


Fig. 2. Blueprint of the NITOS testbed architecture. NITOS testbed's Control, OpenFlow, Chassis Manager, and Wireless Experimental Networks are accessed by remote users via the NITOS server. OMF and NITOS scheduler are software components residing on the NITOS server.

Once granted access to his/her reserved resources, the user has full control over them. However, acknowledging the need to provide to its users an efficient way to handle simple, yet frustrating, management and experiment setup tasks, NITOS testbed has adopted the cOntrol & Management Framework (OMF) [8], already since the beginning of the testbed lifetime. This choice was dictated by the fact that OMF was at the time (and still is) the control and management framework (CMF) with the widest diffusion among testbeds internationally, especially wireless testbeds, so it had been already successively tested by a large user community. OMF, initially developed at the ORBIT testbed [9], but currently maintained and further developed by NICTA [10], allows researchers to describe, instrument and execute their experiments. From a testbed operator's point of view, it provides a set of services to efficiently manage and operate the testbed resources (e.g. installing OS images, resetting nodes).

Apart from these software tools, NITOS features a set of tools related to measurements. The most important is OMF Measurement Library (OML) [11], a framework dedicated to experiment measurements handling, developed by NICTA. Apart from OML, a number of custom applications, developed by UTH, are available for NITOS users. All these tools are described in detail in sections 3 and 4, along with background for their necessity and impact.

Details on how to use the aforementioned software tools in NITOS are provided in the testbed website. A new user is encouraged to start with consulting the *Basic Tutorial* section, available in [12] and then proceed with the rest of the available documentation.

3 Measurements for Topology Assessment

Running a network experiment on a entirely private wired testbed offers the advantage of knowing the capacity of each network link in advance. In a wireless testbed, however, such knowledge can not be safely assumed due to the inherent volatility of the environment. Variance of link capacities is greatly dependent on the specific environment where the wireless testbed is deployed. In the case of Wi-Fi testbeds, we can in general discriminate among the following categories:

- **Indoor deployments in special interference-isolated environments**
  These deployments offer the most stable wireless link capacities. No external sources in the Wi-Fi band operate in the vicinity of the testbed. The link capacities can be considered practically stable over time.
- **Indoor deployments near or colocated with interference sources**
  This is perhaps the most common case, where wireless nodes are located in offices and laboratories in campus or research institute buildings, coexisting with Wi-Fi access points used for Internet access and possibly with other devices operating in the same band (Bluetooth, cordless phones, etc). Interference is a major issue in these deployments, especially during working hours, where external activity in these frequencies reaches its peak.
- **Entirely or partly outdoor deployments in external interference constrained environments**
  These deployments are extremely vulnerable to external interference, but are also affected by weather and atmosphere conditions, due to the effect of varying density of scatterer particles. Changes in the topology of physical obstructions can also affect link capacities, especially in crowded outdoor environments.
- **Entirely or partly outdoor deployments in interference free environments**
  These testbeds are only constrained by fluctuations in the environment, due to weather and atmosphere changing conditions, or due to moving physical obstacles.

The fluctuations in the capacities of the links affect the properties of a given node topology. Typically, a researcher intending to run a wireless experiment selects a subset of resources among the available, based on connectivity between node pairs and the qualities of the different links. Therefore, there is a need for software tools to provide this information to testbed users.

In NITOS there are two software frameworks used in order to assist users in assessing the testbed's wireless environment properties and in selecting an appropriate topology for their experiment. The first one is the *NITOS Topology & Connectivity Tool*, which estimates each link's quality by calculating the Packet Delivery Ratio (PDR) over all combinations of physical transmission rates and frequency channels. The second one is called *NITOS Channel Sensing Framework* [13] and it is based on software-defined radio (SDR) devices that feature highly flexible wireless transceivers and are able to provide highly accurate channel sensing measurements.

3.1 The NITOS Topology and Connectivity Tool

The NITOS Topology & Connectivity Tool [14] collects information for link quality measurements with the aid of *Topology and Link Quality Assessment Protocol* (TLQAP) [15], a framework implemented in the Click modular router [16], taking advantage of the available Click extensions for the MadWiFi driver [17].

TLQAP is based on actual throughput measurements of a fixed number of consecutive packet transmissions, initiated at each testbed node. Each TLQAP session is comprised of a number of fixed size packets which are transmitted in one burst at a specific channel and physical rate combination. These packets are addressed to an arbitrary neighbor node of the current transmitter and are transmitted without 802.11 support for low level acknowledgements and retransmissions. Otherwise, the captured packet loss ratio would be lower than the underlying, actual loss ratio.

At the same time, each non-transmitting node sniffs (in monitor mode) and logs all the TLQAP packets that it can hear. TLQAP packets feature a special header placed immediately after the Ethernet header. The header fields are the sender IP address along with globally agreed identification numbers for the channel and rate that have been used in the current packet transmission. Such frame header modifications were made possible via the use of the MadWiFi open-source driver.

It must be noted that this mechanism is somewhat equivalent to a typical broadcast transmission. However, in 802.11, broadcast transmissions always use the lowest physical rate. To assess link qualities at higher rates, consecutive unicast transmissions would be required, which would undoubtedly greatly increase the complexity and delay of the framework. Thus, through TLQAP, there is a large gain in efficiency and speed.

The PDR from node X to node Y is calculated by dividing the number of packets received by Y by the number sent by X. Originally, the respective session transmission delay was also recorded, in order to assess congestion at each channel due to transmissions from testbed external Wi-Fi transmitters. Currently, however, congestion assessment measurements are being conducted through another framework, described in the next subsection.

The NITOS Topology & Connectivity tool, developed for NITOS in order to exhibit measurements obtained through TLQAP sessions, is comprised of a web interface, a database and a set of .dot scripts. The experimenter can use the interactive web interface to select particular testbed nodes and it in turn provides him/her with a graphical illustration of the outgoing links in the vicinity of each node and the respective PDR measurements. A snapshot of this interface can be viewed in Fig. 3. The PDR measurement data is stored in a database and collected when TLQAP is enabled by the system administration. TLQAP sessions are scheduled frequently, at periods when none of the nodes is reserved, as a means to keep the link quality information up-to-date. The .dot scripts are then used to generate the graphs showing the PDR metrics by submitting queries on the database. In Fig. 4, the quality of links from node 1

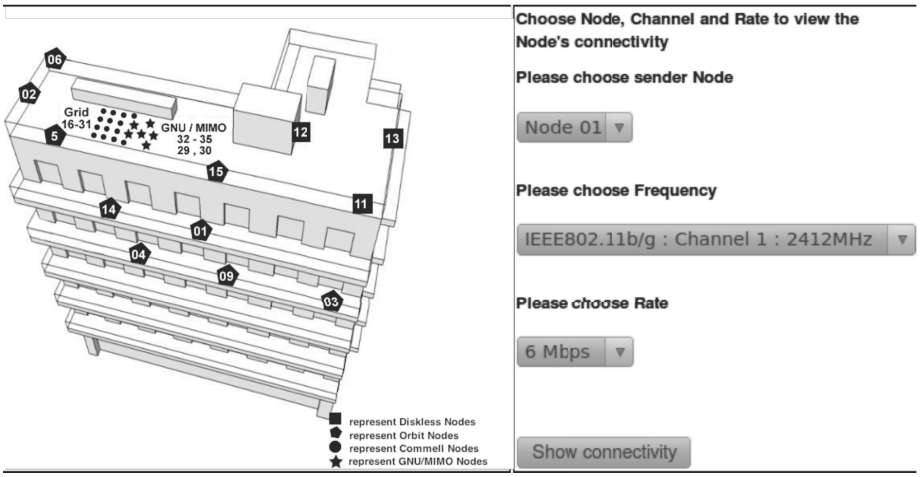


Fig. 3. Web interface for connectivity tool

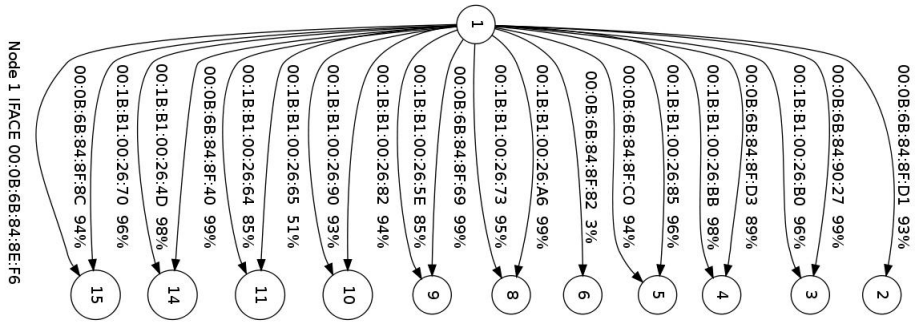


Fig. 4. Link quality graph for node 1

towards its neighbor nodes is illustrated. Each node is indicated by a circle and the PDR of each link is reported on graph edges that indicate link connectivity towards Wi-Fi interfaces of certain nodes.

3.2 The NITOS Channel Sensing Framework

Link quality information is often not sufficient on its own, as congestion and interference from external devices might significantly degrade link throughput. These effects could be caused by external Wi-Fi sources, such as access-points in the vicinity, or non-802.11 devices operating in the unlicensed band, such as Bluetooth devices or microwave ovens. It is important to note that if the signal

received by a node is not 802.11-compliant, the receiver will not be able to decode it. Furthermore, if the received signal level is below the channel sensing threshold of a node's wireless card, the node will not back off when it has a frame to transmit. For these reasons, using packet sniffers to detect traffic or relying on the 802.11 backoff mechanism to assess it are not totally reliable methods, and a more sensitive mechanism should be used instead.

Since NITOS is an outdoor-deployed, non-RF-isolated testbed with significant external interference conditions, it is important to provide its users with a powerful spectrum sensing platform, that must be able to exhibit accurate information regarding the 802.11 and non-802.11 activity in each Wi-Fi band channel. For this purpose, the NITOS Channel Sensing Framework [13] is used, which exploits the spectrum sensing capabilities of the NITOS testbed's Software Defined Radio boards. In particular, the used devices are the computer-hosted USRP boards [18] from the Ettus company. Such a board can be viewed mounted on a NITOS node in Fig. 5. These are used in conjunction with GNU Radio [19], an open-source software development toolkit that provides signal processing blocks to implement software radios. A total number of 9 wireless nodes are equipped with USRP1 and USRP N210 devices, spanning the NITOS testbed deployment.

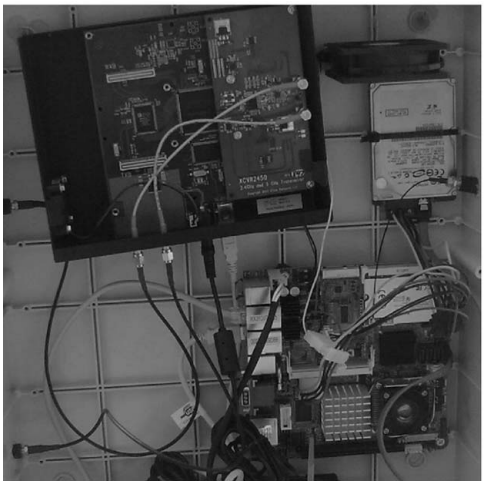


Fig. 5. USRP board mounted on NITOS node

Focusing on the spectrum sensing goal, the Channel Sensing Framework estimates the occupancy ratio per sampled frequency, considering the Received Signal Strength (RSS) measurements that exceed a predefined RSS threshold.

Further details for Channel Occupancy Ratio computation can be found in [20]. The framework allows a user to enable scripts that configure a multitude of sampling parameters, such as:

- the list of frequencies that will be sampled,
- the duration of sampling per individual frequency,
- the number of iterations of the repeated sensing procedure,
- the overall sampling period,
- and the RSS threshold that will be used for measurement filtering.

A flowchart representation of the underlying sensing algorithm is depicted in Fig 6 and shows the execution steps of the spectrum sensing procedure, as triggered by the input configuration parameters provided by the user.

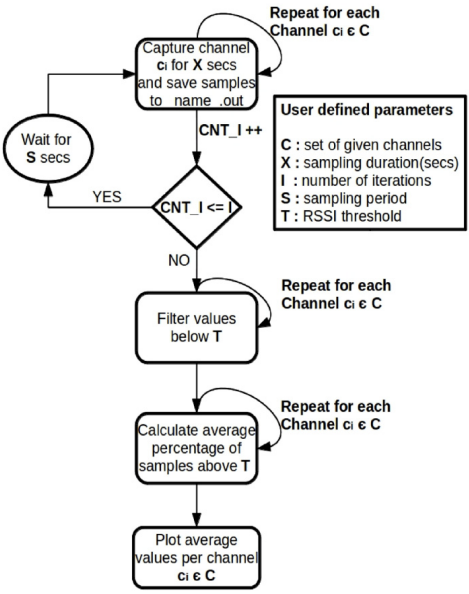


Fig. 6. Flowchart of the Channel Sensing Framework Procedure

The user is able to get a graphical representation of each measurement set that has been already stored in NITOS database, through a web interface. An example is illustrated in Fig. 7, where we can see channel occupancy for the 2.4GHz and 5GHz bands. Various statistical measures can be extracted from the corresponding records, such as average and deviation values per frequency or per individual iteration, and correlation data for measurements captured by different USRP-enabled nodes.

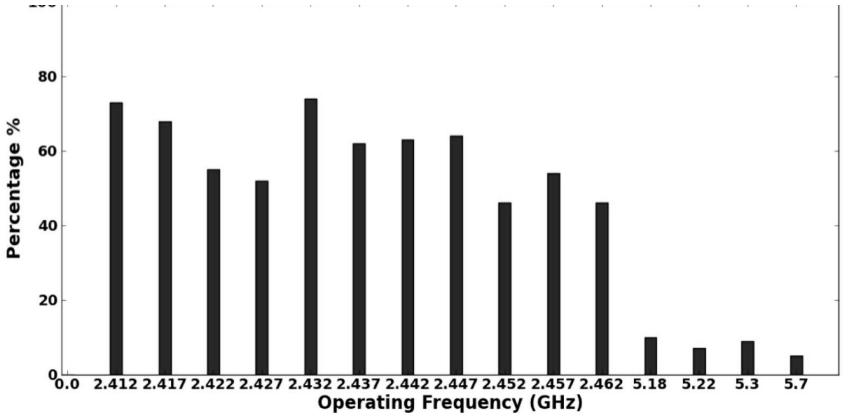


Fig. 7. Plot of the COT measurements per selected frequencies (in 2.4 GHz & 5 GHz bands)

4 Measurements for Analysis of Experimental Results and Experiment Steering

Selection of an appropriate topology for an experiment is only the first step for the experimentation procedure. The most important part of the procedure is, however, collection of measurements at experiment runtime. This constitutes in capturing the values of specific experiment variables emanating from typically multiple sources in a synchronized and organized manner. These variables may be generated in software applications running in some of the experiment's resources, or may be capturable measurements of interest, which assist the user in evaluating the experiment's outcome.

The basic framework used in NITOS for handling experiment measurements is OML [11], which stands for OMF Measurement Library. OML was initially developed as part of the OMF testbed control and management framework, but it evolved in an independent software. Both OMF and OML are being developed by NICTA (Sydney, Australia) as free and open source software. Another tool used in NITOS, but currently still restricted for developers, is spectrum monitoring during an experiment, through the USRP-enabled nodes. Below we describe these components in detail.

4.1 The OML Measurement Framework

OML is a framework designed to support the entire lifecycle of measurements, from their generation and capturing, to their storage and visualization. In Fig. 8 we can see an overview of the framework's architecture. It is based on the client-server paradigm, where an OML server collects the measurements from client applications, running on experiment resources.

Measurements are captured at the resources in so-called measurement points, which are pieces of code which inject given variables to the OML server. OML provides dedicated software libraries which allow insertion of such code inside the source code of the applications or building of wrapper OML-enabled applications around the interface of the original ones. Measurements may pass through a sequence of filters before sent to the server, if the user wishes to perform some processing after their capturing. Examples of available filters are averaging over a given time interval, or selecting the maximum value among a fixed number of measurements.

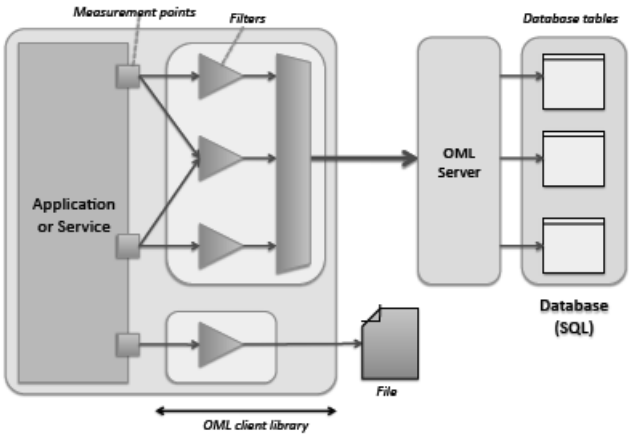


Fig. 8. Overview of OML architecture (source: http://mytestbed.net)

At the server side, measurements are stored in Sqlite databases, organized with timestamps. In an OMF experiment, all the measurements are gathered in a single database corresponding to the experiment, and measurements from different applications are stored in different tables of this database.

Another useful functionality offered by OML is the ability to visualize graphical representations of specific variables at experiment runtime. In this way, the researcher can save a lot of time in the process of evaluating the outcome of an experiment, and perhaps even stop execution if he/she realizes that it does not evolve as expected. In Fig. 9 a snapshot of such a graphical representation can be viewed.

An important featured offered by OML, in conjunction with OMF, is the ability to buffer measurements inside a resource and dump them to the OML server in a batch. This is extremely useful in cases involving mobile resources, where the control network is wireless and connectivity may be temporarily lost between a resource and the server for a particular period. Originally, switching between local buffering of data and sending them to the server was handled manually. UTH contributed an improved version of this feature, where a daemon checking connectivity between the resource and the server is in control of the operating mode.

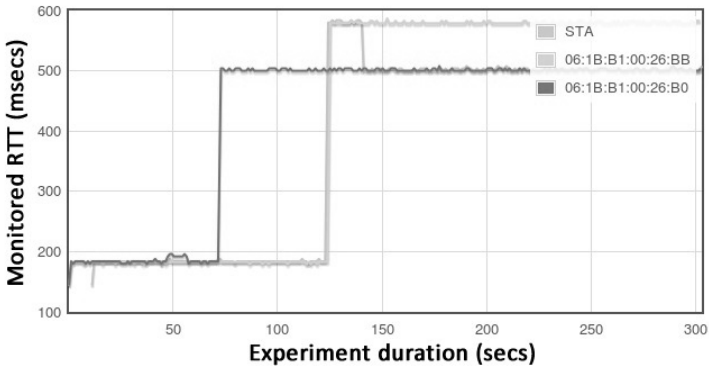


Fig. 9. Runtime visualization of experiment measurements through OML

4.2 Measurement-Based Experiment Steering

As already mentioned, OML was originally a companion framework for OMF. Therefore, it is not surprising that the OMF experiment control framework features various functionalities that exploit the capabilities of OML. Perhaps the most interesting from an experimenter's point of view is the ability to steer an experiment based on the values of specific measurements.

This feature is part of OMF's event-based mechanism. With this mechanism, one can define specific events in an experiment description script, and specific actions to take place when these events are triggered. An example of such a scenario would be for example a dynamic re-association of a wireless station to a new access-point if it experiences congestion over a given level, where the level of congestion is measured and updated dynamically through OML. If all access-points in the area exceed that threshold, then the station could just associate to the access-point experiencing the least congestion (i.e. the trigger condition could be comparison between measurements).

4.3 Spectrum Activity Monitoring

Apart from providing an accurate image of channel occupancy and activity before the execution of an experiment, which serves as a preventive measure against experiencing severe interference, a wireless testbed should ideally offer the ability to experimenters to monitor activity during experiment runtime, particularly activity in the channels utilized in the experiment. This allows to match an unexpected fluctuation in the performance metrics of an experiment with an associated traffic burst emanating from an external source, causing interference or congestion. For instance, a link's throughput could drop to a half of what it was, if during an experiment an external link in the vicinity is using the same channel for data transfer.

NITOS is looking to fill that gap through a more sophisticated framework around its software defined radio enabled nodes. This is currently in development

stage, and still not offered to users of the testbed. The idea is that in parallel with an OMF experiment, a background monitoring of the experiment's frequencies takes place from USRPs in the vicinity of the nodes. The user can then compare experiment measurements with spectrum activity data, possibly even by visualizing them in a single graph window.

5 Examples of Experiments Conducted in NITOS

In this section, we describe two experiments conducted in the NITOS testbed, as part of the research activities in European projects which utilize its services. With each description, we also explain the tools used for measurements, so as to provide examples of their usage in real experiments.

5.1 The CONECT Project's Diamond Network Experiment

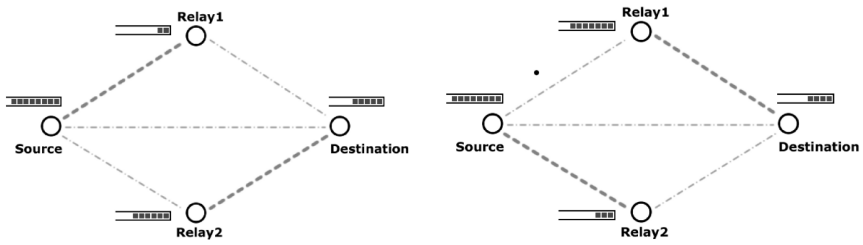
The Cooperative Networking for High Capacity Transport Architectures (CONECT) [21] project proposes a holistic network design approach, by exploiting cooperative forwarding strategies. In order to harvest the benefits of such a research effort and validate the designed cooperative protocols, NITOS is used as a performance evaluation tool that ensures the validity and robustness of the cooperative schemes proposed.

Briefly, the work being done in CONECT involving NITOS includes sending unicast traffic through a proposed cooperative relaying scheme, as well as an extensible scenario over multicast sessions. It also includes insights for applying those cooperative schemes to realistic scenarios, where the benefits of collaboration among nodes can reveal the prospective performance improvements.

The relay-assisted topology in Fig. 10 illustrates the canonical diamond network, containing a source, two relays and a destination (single unicast session), where we consider the joint scheduling and power allocation problem with the objective of stabilizing the network and either maximizing throughput or minimizing total power consumption [21],[22]. There is a network controller that chooses between two feasible scheduling action sets, activating either the set depicted in Fig. 10(a) or the one in Fig. 10(b). The transmissions permitted at each time slot are indicated with bold dotted lines. The information flow is towards the destination, and since the networking conditions (buffer queuing, channel quality) change over time, this will cause the network controller that implements an optimal decision policy to select a particular schedule. The key principle is that the scheduling decision process does not lead in system starvation, since changes in network dynamics are coupled with schedule selection [22].

In Fig. 11 the multicast extension of the problem is depicted. Enhancing the previous scenario of activating single session schedules, to improve networking performance, we consider an extension of the cooperative notion in multicast wireless scenarios, where we assume the existence of a multicast group that desires to experience QoS guarantees.

All these scenarios require careful design in the experimentation methodology and in the evaluation of the results. Instrumentation and repeatability in experimentation are of vital importance for collecting rigorous results and evaluating the performance of the cooperative schemes. An ideal feature would be the ability to exactly match the gathered experimentation results with those being proved by the use of mathematical tools and theory [22]. However, this is not feasible, not only because of the gap between theory and practice caused by modeling simplifications that theory adopts, but also because of a potential lack in a systematic method of experimentation. With the joint use of the OMF framework and the NITOS tools (Connectivity Tool, Scheduler) [7,14,15], reproducibility of the experiment flow is achieved along with monitoring of the varying channel conditions per run. These imply multiple benefits for the experimenter, such as the ability to evaluate the robustness of an algorithm against channel conditions or against different topologies, test the accuracy of the model adopted in theoretical analysis regarding the wireless environment, and derive average and outage performance metrics through repeated experiments.



(a) 1st feasible scheduling set (b) 2nd feasible scheduling set

Fig. 10. Enabling unicast parallel transmissions in the diamond network. Each subfigure indicates a feasible schedule that can be activated each time slot.

In Fig. 12 we can see an example of a diamond network topology in NITOS. It shows some nodes that can be selected to form a diamond network in the NITOS grid. This pattern is not limited to the particular nodes, but it can be extended and repeated among all combinations of nodes forming canonical diamonds, and span over the spatial imprint of the NITOS testbed grid. In order to select topologies that best fit on the relaying scenario objective, a user can exploit the NITOS Connectivity tool that was previously introduced. In this way, an *a priori estimate* of the networking states is given, so as to evaluate the gathered performance measurements.

Collection of measurements takes place with use of the OML framework. The forwarding mechanism is developed inside the IP/MAC layer driver, however the procedure of collecting valuable measurements regarding per packet power consumption, rate configuration and throughput is controlled and configured through OMF/OML. For the sake of evaluating the versatility of the experiments, the experiments were run multiple times, with USRPs sensing the related

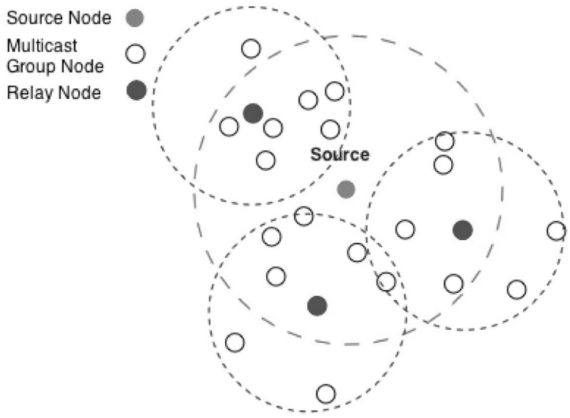


Fig. 11. A source node is aided by some relays that they forward the traffic to the members of the multicast group

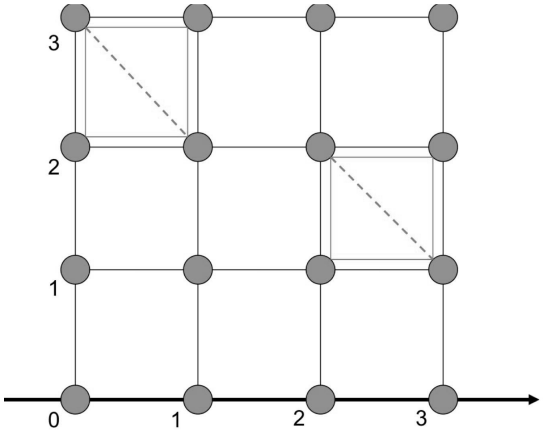


Fig. 12. A diamond network mapping onto the NITOS grid

frequencies in the background. Whenever an unexpected pattern occurred in one of the experiment instances, we checked against the spectrum data for unusual activity bursts and, in case of a positive match, this instance was excluded from the results taken into account.

5.2 Vehicle to Infrastructure (V2I) Communication Network

In the context of the European project REDUCTION [23], whose objective is to combine vehicular and Information & Communication Technologies (ICT) for

reducing the environmental footprint of vehicles monitored by multi-modal fleet management systems, an experiment involving a Vehicle to Infrastructure (V2I) communication network was conducted. The main idea of the experiment was to demonstrate the scenario where a vehicle equipped with a set of sensors gathers measurements from its environment and communicates opportunistically with Road-Side-Units (RSU), in order to forward the measurements to a centralized framework for storage and analysis.

The connection between the car-mounted node and the RSU was achieved through a Wi-Fi interface. The communication protocol used for this set up was 802.11p, which implies operation in the 5.9 GHz band and use of 10 MHz channel bandwidth (instead of 20MHz used in 802.11a/b/g).

A static NITOS node was used as the RSU, while another NITOS node, mounted on a vehicle (car) was used for gathering measurements and forwarding them to the RSU. A set of sensors was connected to the vehicle-mounted node, sensing temperature, humidity and CO_2. The sensors were connected to an Arduino Uno [24] board attached to the node (cf. Fig. 13). Additionally, a GPS module was connected to the mobile node, enabling measurement of latitude, longitude, altitude, speed, vertical speed and direction.

(a) Arduino board (b) CO_2 sensor (c) Temperature and humidity sensor

Fig. 13. Sensors used on vehicle-mounted node

For the purpose of collecting the measurements, an OML-enabled application was developed for each one of the three types of sensors. Furthermore, the OML disconnected experiment feature was exploited, as the mobile node moved out of the OML server's connectivity range for restricted periods. In order to switch between disconnected and connected mode, the enhanced version with the connectivity monitor daemon developed by the UTH team was used.

In Fig. 14 we can see an overview of the experiment, including a snapshot of the web interface that was created for demonstration purposes, based on the Google Maps API [25]. The data depicted at each point in the map was extracted by the experiment's Sqlite database (in the NITOS server), after conversion into an XML file.

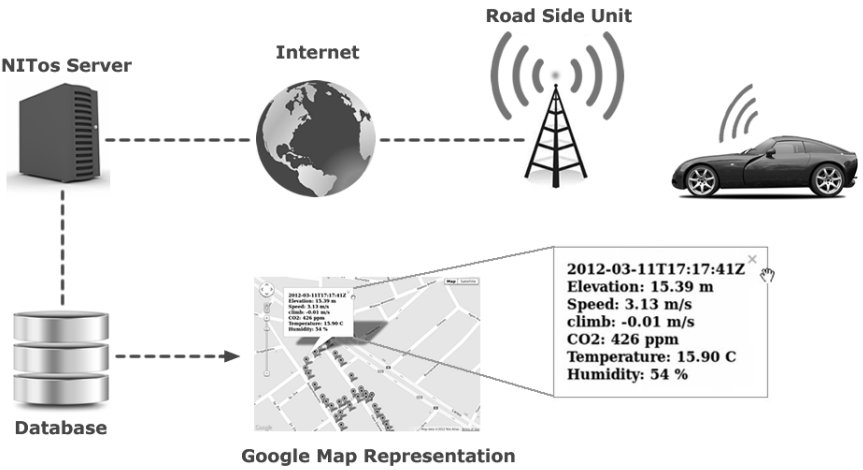


Fig. 14. Overview of V2I demo experiment

6 Challenges - Future Plans

There are a lot of challenges, particularly related to the nature of the wireless environment, which complicate the development of efficient measurement tools for wireless testbeds. We describe some below, along with future plans of NITOS to address them.

As different testbed nodes are affected from different interference sources, depending on their location, channel sensing information should be available on a per-node level. This can be approximated, by spreading the available USRP boards uniformly across the testbed area. Then, for each node, the measurement data from USRPs in its vicinity should be correlated, in order to produce a good estimate of the actual interference perceived by the particular node. Accuracy of the estimates depends of course on the density of USRPs, so there is a tradeoff between the deployment cost of USRPs and the quality of estimation.

Another challenge is related with correlation of experiment results with external interference measurements during an experiment. This is in large part a technical task, involving synchronization between different measurement streams. It is in the plans of NITOS to integrate such a functionality in its measurement toolkit, as already discussed, and preliminary tests are taking place.

Looking at wireless topologies from an experimenter's point of view, further challenges arise. An experimenter might expect the testbed tools to propose him/her an appropriate topology for a specific experiment, instead of using the existing tools to find it out on his/her own. That is, intelligence should be added to the existing topology assessment measurement tools. For instance, an experimenter would just want a "working" diamond topology in the diamond network experiment, not caring about which specific nodes he would be proposed. On the other side, the range of potential experiment setups is so large, that it is not

always straightforward to describe the best suited topology efficiently. In case of multiple simultaneous experiments, a combinatorial optimization problem might have to be solved. UTH's group working at NITOS is investigating these issues, with the intention to find the best practical solution possible.

Finally, development of measurement and benchmarking tools for WiMAX and LTE, especially in the form of mobile applications, is an ongoing task. These tools are planned for use in conjunction with the WiMAX and LTE testbed facilities that will be added to NITOS.

7 Conclusions

As measurements constitute the most sensitive part of the experimentation procedure, there is a need for a set of tools and methodologies to efficiently handle related testbed functionalities. In wireless testbeds, where the uncertain environment may affect experiment results, building such a set of tools is particularly challenging. We presented tools and methodologies used by the NITOS testbed in its effort to address these issues, explaining the utility and limitations of each of them. We also provided examples of real experiments, conducted in the context of European research projects, where these instruments were put in practice.

References

1. Feldmann, A.: Internet clean-slate design: what and why? ACM SIGCOMM Computer Communication Review 37, 59–64 (2007)
2. FIRE, Future Internet Research & Experimentation, http://cordis.europa.eu/fp7/ict/fire/home_en.html
3. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: OpenFlow: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. 38(2), 69–74 (2008)
4. OneLab, Future Internet Testbeds, https://onelab.eu/
5. OpenLab, http://www.ict-openlab.eu/
6. Sherwood, R., Gibb, G., McKeown, N., Parulkar, G.: Flowvisor: A network virtualization layer. Tech. Rep. OPENFLOW-TR-2009-1, Deutsche Telekom Inc. R&D Lab, Stanford University, Nicira Networks (October 2009)
7. The NITOS Scheduler Management Tool, http://nitlab.inf.uth.gr/NITlab/index.php/scheduler
8. Rakotoarivelo, T., Ott, M., Jourjon, G., Seskar, I.: OMF: a control and management framework for networking testbeds. SIGOPS Oper. Syst. Rev. 43, 54–59 (2010)
9. ORBIT Wireless Network Testbed, http://www.orbit-lab.org/
10. NICTA, National ICT Australia, http://www.nicta.com.au/
11. White, J., Jourjon, G., Rakatoarivelo, T., Ott, M.: Measurement Architectures for Network Experiments with Disconnected Mobile Nodes. In: Magedanz, T., Gavras, A., Thanh, N.H., Chase, J.S. (eds.) TridentCom 2010. LNICST, vol. 46, pp. 315–330. Springer, Heidelberg (2011)
12. Basic Tutorial for using NITOS, http://nitlab.inf.uth.gr/NITlab/index.php/testbed/instructions/basic-tutorial

13. Passas, V., Keranidis, S., Korakis, T., Koutsopoulos, I., Tassiulas, L.: An Experimental Framework for Channel Sensing through USRP/GNU Radios. In: Korakis, T., Zink, M., Ott, M. (eds.) TridentCom 2012. LNICST, vol. 44, pp. 383–386. Springer, Heidelberg (2012)
14. Apostolaras, A., Miliotis, V., Giallelis, N., Syrivelis, D., Korakis, T., Tassiulas, L.: A Demonstration of a Management Tool for Assessing Channel Quality Information in Wireless Testbeds. In: Magedanz, T., Gavras, A., Thanh, N.H., Chase, J.S. (eds.) TridentCom 2010. LNICST, vol. 46, pp. 615–618. Springer, Heidelberg (2011)
15. Syrivelis, D., Anadiotis, A.-C., Apostolaras, A., Korakis, T., Tassiulas, L.: TLQAP: a topology and link quality assessment protocol for efficient node allocation on wireless testbeds. In: Proceedings of the 4th ACM International Workshop on Experimental Evaluation and Characterization, WINTECH 2009, pp. 27–34. ACM, New York (2009)
16. Morris, R., Kohler, E., Jannotti, J., Kaashoek, M.F.: The Click modular router. SIGOPS Oper. Syst. Rev. 33, 217–231 (1999)
17. The MadWiFi project, `http://madwifi-project.org/`
18. The Ettus Universal Software Radio Peripheral, USRP, `https://www.ettus.com/product/`
19. The GNU Radio, `http://gnuradio.org/`
20. Kazdaridis, G., Keranidis, S., Fiamegkos, A., Korakis, T., Koutsopoulos, I., Tassiulas, L.: Novel metrics and experimentation insights for dynamic frequency selection in wireless LANs. In: Proceedings of the 6th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, WiNTECH 2011, pp. 51–58. ACM, New York (2011)
21. CONECT-Cooperative Networking for High Capacity Transport Architectures, `http://www.conect-ict.eu/`
22. Apostolaras, A., Cottatellucci, L., Gatzianas, M., Koutsopoulos, I., Li, Q., Navid, N., Wang, L.: D2.2: Advances on packet-level cooperation techniques for unicast traffic transmission. CONECT Project Deliverable (2011), `http://www.conect-ict.eu/`
23. REDUCTION: Reducing Environmental Footprint based on Multi-Modal Fleet management Systems for Eco-Routing and Driver Behaviour Adaptation, `http://www.reduction-project.eu/`
24. The Arduino Uno board, `http://arduino.cc/en/Main/arduinoBoardUno/`
25. The Google Maps API, `https://developers.google.com/maps/`