Πανεπιστήμιο Θεσσαλίας, 2014-2015

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Διπλωματική Εργασία

Θέμα:

«Ανάπτυξη διαδικτυακής πύλης πειραματισμού με την ερευνητική υποδομή του NITOS»

«Design and development of the NITOS portal for experimentation with NITOS facility»

Ζαμίχος Αλέξανδρος

UNIVERSITY OF THESSALY

Επιβλέπων Καθηγητής:

Κοράκης Αθανάσιος (Επίκουρος Καθηγητής)

Συν επιβλέπων Καθηγητής:

Αργυρίου Αντώνιος (Λέκτορας Καθηγητής)

Βόλος, Μάρτιος 2015

CONTENTS

CONTENTS 2

Ευχαριστίες 3

ΠΕΡΙΛΗΨΗ 4

ABSTRACT 5

1. Introduction 6

  1.1 Testbed 6

  1.2 NITOS facility 7

    1.2.1 Network Experimentation 7

    1.2.2 Switches 12

    1.2.3 CM cards 13

    1.2.4 USRPs equipment 14

    1.2.5 Enabling WiMAX connectivity to the NITOS testbed 14

    1.2.5 LTE testbed 15

    1.2.6 Facility architecture 16

    1.2.7 Broker 18

  1.3 Motivation 20

2. Tools 21

  2.1 Ruby on Rails 21

  2.2 Structure and Style 21

  2.3 Gems 22

3. Development of the Portal 23

  3.1 Design Database and Log in/out Mechanism 23

  3.2 NITOS Portal Architecture 27

  3.3 Experimentation Tools 28

    3.3.1 User's Timezone Detection 28

    3.3.2 Scheduler Tools 30

    3.3.3 My Reservations Tool 42

    3.3.4 Node Status tool 43

    3.3.5 Other Features 45

Conclusion 48

References 49

Ευχαριστίες

Με την εκπόνηση της παρούσας Διπλωματικής εργασίας, φέρνω εις πέρας τις προπτυχιακές μου σπουδές στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

Θα ήθελα αρχικά να ευχαριστήσω θερμά τον κ. Κοράκη Αθανάσιο, Επίκουρο Καθηγητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, για την ανάθεση της Διπλωματικής μου εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω ιδιαιτέρως τους Νιαβή Χάρη και Δαδούκη Άρη, ερευνητές του Πανεπιστημίου Θεσσαλίας, για την συνεχή καθοδήγηση και ουσιαστική βοήθειά τους, για την ολοκλήρωση της Διπλωματικής μου εργασίας.

Ακόμα, ευχαριστώ την κοπέλα μου και όλους τους φίλους μου, για την υποστήριξη και την συμπαράστασή τους καθ' όλη τη διάρκεια των σπουδών μου.

Τέλος, ευχαριστώ θερμά την οικογένειά μου για τη βοήθεια, τα εφόδια και την αμέριστη συμπαράσταση που μου παρείχε όλα αυτά τα χρόνια.

Στην οικογένειά μου

ΠΕΡΙΛΗΨΗ

Η παρούσα Διπλωματική Εργασία ασχολείται με την υλοποίηση ενός πειραματικού Portal για την ερευνητική εγκατάσταση NITOS.

Το πρώτο μέρος της εργασίας περιέχει μια γενική εισαγωγή για τα testbeds και μια πιο ειδική για το testbed NITOS, αλλά και το κίνητρο για την υλοποίηση του πειραματικού Portal.

Στο δεύτερο μέρος παραθέτουμε τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη του Portal και τους λόγους που επιλέχθηκαν.

Στο τρίτο μέρος βρίσκεται η υλοποίηση του Portal. Αρχικά παρουσιάζουμε την αρχιτεκτονική του και στη συνέχεια αναλύουμε κάθε κομμάτι του ξεχωριστά. Τέλος, δίνουμε μερικά screen shots του Portal.

ABSTRACT

The present Final Project Dissertation – Thesis is an implementation of an Experimental Portal for the NITOS research facility.

The first part of the project, consists of an introduction about testbeds in general and more specifically about NITOS testbed and the motivation for implementing this Experimental Portal.

At the second part, we present the tools that we used for the development of the Portal and the reasons that we used them.

At the third part lies the implementation of the Portal. Firstly we present the architecture of the Portal and then we resume with a detailed presentation of every component of the Portal. Finally we give some screen shots of the Portal UI.

1. Introduction

1.1 Testbed

Wireless networking experimentation is commonly used for protocols benchmarking and validation. In this direction, experimentation platforms have created, commonly known as testbeds. The aim of testbeds is to provide means for the experimentation definition and performance evaluation.

A testbed is a platform for experimentation of many different projects. Testbeds allows the research community to test and evaluate the scientific theories, new technologies and applications. Testbeds are a critical tool for evaluating ongoing development.

Research in wireless networks include theoretical analysis as also experimentation. And although the theoretical analysis is the key to efficient solutions, experimentation will confirm the results to real world scenarios.

A common way to perform wireless experimentation is by using simulations. Various simulators have been developed such as NS-3 and GloMoSim. While such simulators can define the exact environment where the theoretical model will be testing, they cannot emulate the phenomena that may take place in real world. As an example, simulators include inaccurate representation of the wireless medium and ignore several aspects such as computational overhead.

Due to this limitations, real platforms preferred from researches for testing the theoretical analysis. Indeed, use of testbeds in networking as evaluation platforms has steadily increased over the last years, something that occurs from the increased number of testbed development.

Moving from simulation to real testbed experimentation more realism rather than abstraction is experienced, while also advantages of applicability on real case scenarios are offered. Moreover, while in simulation/emulation, scalability and reproducibility of an experimentation is more likely to be controlled, in a real wireless testbed, a service/framework that can provide means for scalability and reproducibility becomes more enticing for offering advanced instrumentation features.

The latest trend in testbed's community is federating between testbeds, in Europe and all over the world. Testbed's community tries to take advantage of the distinct infrastructures, of the numerous wireless testbeds, and provide federated wireless testbeds through the interconnection of various facilities and the adoption of similar tools and methodologies. Giving the opportunity to users to have access to a larger amount of resources and making experiments more complicated.

1.2 NITOS facility

Among the wireless testbeds, Network Implementation Testbed Laboratory (NITlab) of the Computer and Communication Engineering Department at University of Thessaly, has a testbed named NITOS, which stands for *Network Implementation Testbed* using *Open Source* platforms. NITOS testbed currently consists of 100 operational wireless nodes, which are based on commercial Wifi cards and Linux open source drivers. The testbed is designed to achieve reproducibility of experimentation, while also supporting evaluation of protocols and applications in real world settings.

NITOS facility is open to the research community 24/7 and it is remotely accessible through the NITOS reservation tool. Different users can use this platform parallel, through the utilization of the NITOS scheduler software. The testbed is based on open-source software. The control and management of the testbed is done using the cOntrol and Management Framework (OMF) open-source software. Users can perform their experiments by reserving slices (nodes, frequency spectrum) of the testbed through the NITOS scheduler.

1.2.1 Network Experimentation

NITOS facility is comprised of 2 wireless testbeds for experimentation. An outdoor testbed, featuring WiFi, WiMAX and LTE support and an indoor isolated testbed comprised of advanced powerful nodes.

1.2.1.1 Outdoor testbed

NITOS outdoor testbed consists of powerful nodes that feature multiple wireless interfaces and allow for experimentation with heterogeneous (WiFi, WiMAX, LTE) wireless technologies. It is deployed at the exterior of the University of Thessaly (UTH) campus building.

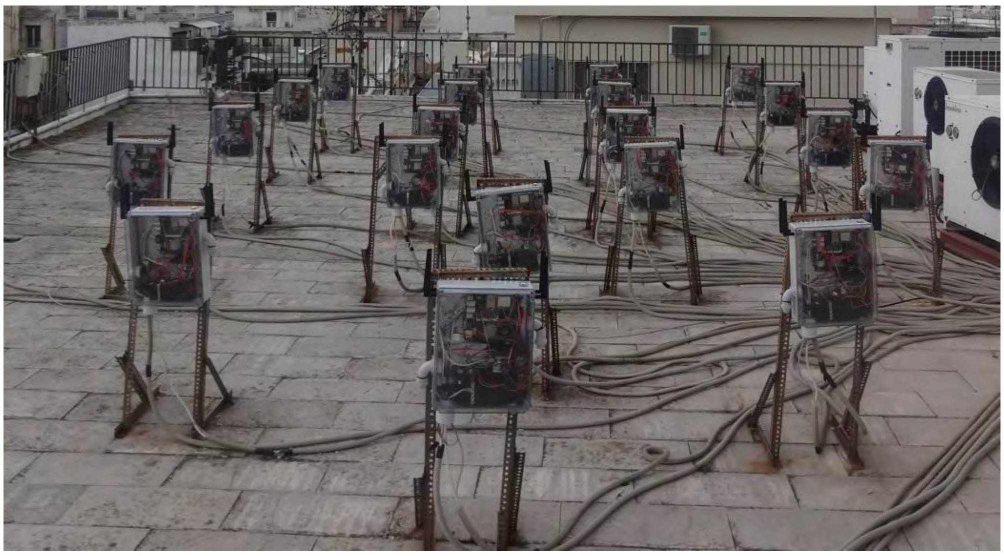


Figure 1: Outdoor testbed

More specifically, it consists of 45 nodes of different types, namely 10 Orbit - like nodes, 20 custom made outdoor nodes (hereafter referred as Outdoor nodes) and 15 diskless nodes.

1.2.1.1.1 Wireless nodes

Orbit – like nodes

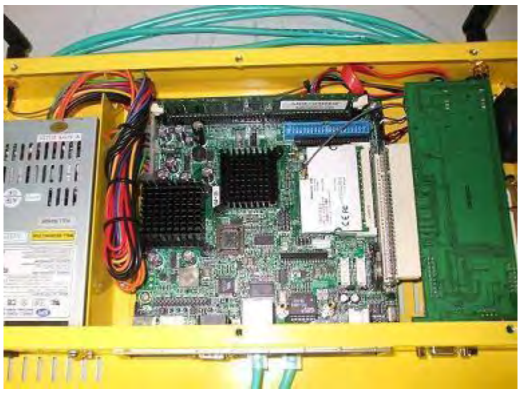


Figure 2: Orbit – like Node

UTH's Outdoor Nodes

The outdoor nodes are equipped with a Chassis Manager (CM) card in order to enable remote control of their operational status.

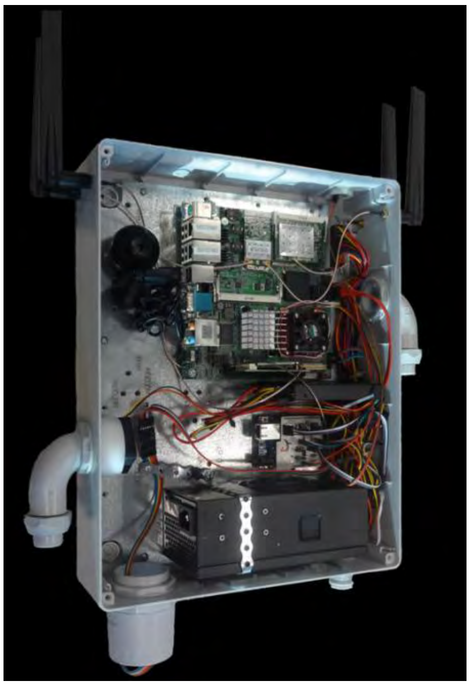


Figure 3: UTH's Outdoor Node

Diskless nodes

The basic characteristic of diskless nodes is the lack of a local disk, due to their low cost nature.

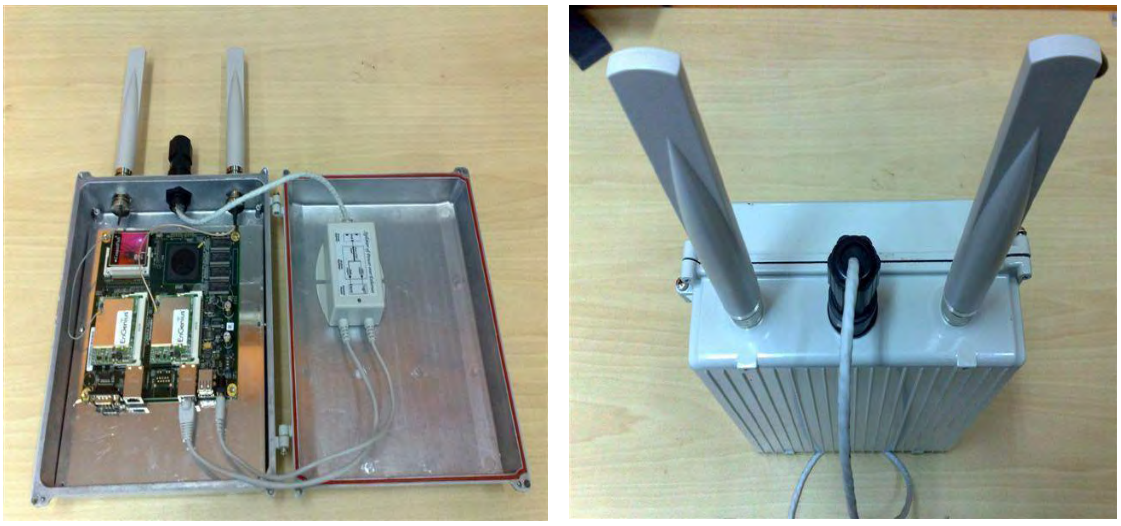


Figure 4: Diskless Node

1.2.1.1.2 Servers and Racks

In NITOS outdoor testbed's server room there are two racks. These are responsible for interconnecting the wireless nodes. The first rack contains the server of the testbed, a UPS and the nodes' plugs connected to the power supply.

The server machine hosts multiple virtual machines, each one standing for a different service providing by the NITOS facility. Namely, the NITOS server for wireless experimentation, the OpenFlow controller and 2 web site servers.

The NITOS server is used to run experiments on the testbed through OMF, and to host various network services including DHCP, DNS, NTP, TFTP, PXE, Frisbee, MySQL, OML and Apache. Furthermore, it is responsible for scheduling of the testbed's resources and for handling users' requests to access the testbed. It also keeps valuable information for experimentation.

The second rack contains all necessary switches of the testbed interconnected with the nodes.

1.2.1.2 Indoor testbed

1.2.1.2.1 Wireless nodes

The new NITOS indoor testbed consists of 40 Icarus nodes and is deployed in an isolated environment at a University of Thessaly's campus building. Experimenters are able to run and evaluate power demanding processing algorithms and protocols in a large scale testbed. It is also equipped with directional antennas.

Icarus Nodes

To control and monitor Icarus node's operation NITOS uses UTH's Chassis Manager Card. A tiny web server is running on CM card and serves http requests such as power on/off and reset commands. Furthermore, the CM card can support real time sensor measures.

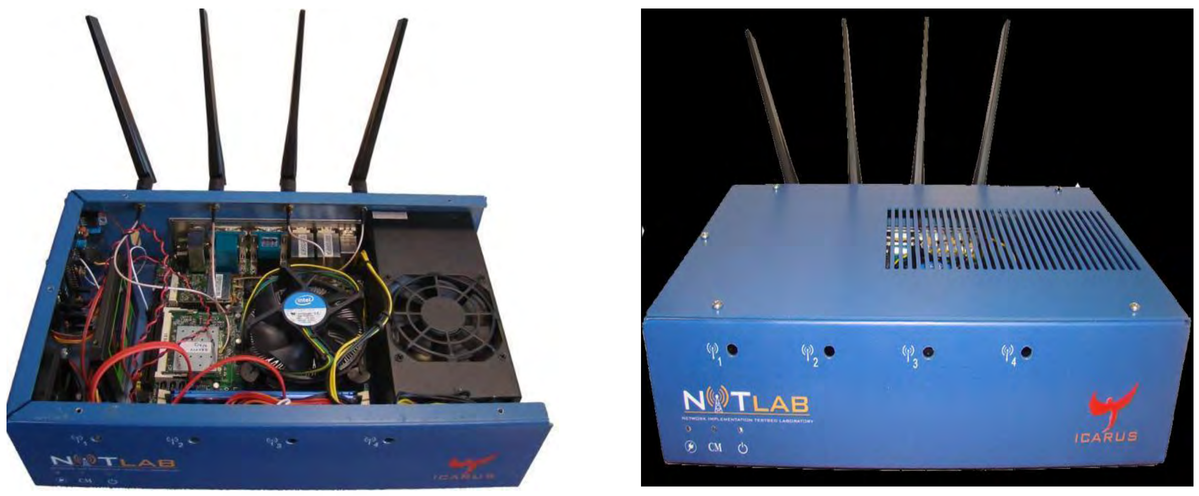


Figure 5: Icarus Node

1.2.1.2.2 Server machine

The NITOS indoor testbed's server machine is a HP ML350p G5

1.2.2 Switches

NITOS testbed operates two HP 2510-48 Switches, in order to enable remote control of the wireless nodes. The first switch is used for sending control messages in order to manage the execution of an experiment through OMF, as well as to collect all measurements defined by the user. The second one is used for interconnecting the CM cards attached on each node, in order to remotely control their operational status.

Figure 6: HP 2510-48 switch

It also features two OpenFlow Ethernet switches – Pronto 3290, in order to interconnect the wireless nodes through a wired network, providing software defined networking (SDN) experimentation to the experimenters.

Figure 7: Pronto 3290

1.2.3 CM cards

UTH has an essential device to control the operational status of NITOS testbed, named Chassis Manager (CM) Cards. The CM cards consists of an Arduino based board and an Arduino compatible shield. The Arduino board includes a general-purpose microcontroller and an Ethernet microcontroller, while the shield includes a couple of relays, a set of sensors and several electronic components. The relays are used to short the power on/off and the reset pins of the node’s motherboard. The sensors provide information about the environmental conditions, such as the temperature the humidity and the light intensity, as well as the internal temperature of the node. The CM card runs a tiny web-server which is responsible for receiving incoming commands via http requests, parsing the commands, making the appropriate actions and responding back to the client with a standard http response.

Part of this thesis was to make a web-based tool that controls and shows the status of the nodes according to user's reservation. Throw this tool user is able to turn on/off reset and see the status of the nodes that has reserved. We discuss node status tool, more detailed, later on this presentation.

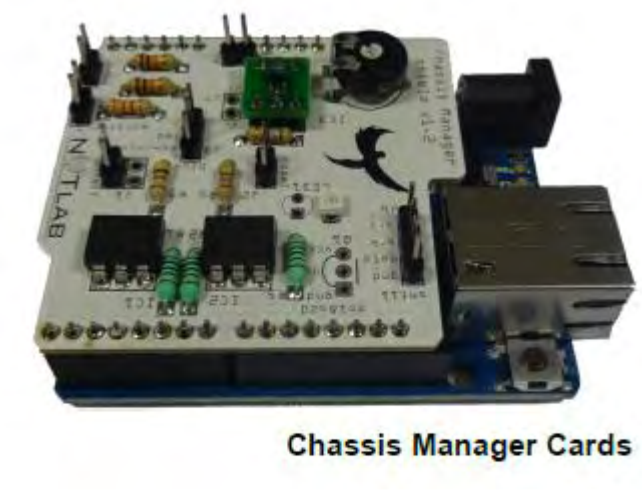


Figure 8: Chassis Manager Cards

1.2.4 USRPs equipment

NITOS has a software defined radio (SDR) testbed that consists of 10 Universal Software Radio Peripheral (USRP) devices attached to the NITOS wireless nodes. USRPs allow the researcher to program a number of physical layer features (e.g. modulation), thereby enabling dedicated PHY layer or cross-layer research.

1.2.5 Enabling WiMAX connectivity to the NITOS testbed

WiMAX (Worldwide Interoperability for Microwave Access) is a wireless communications standard designed to provide 30 to 40 megabit-per-second data rates, and had been adopted as a 4G technology worldwide. NITOS team added a WiMAX Base Station and enabled WiMAX connectivity of the nodes.

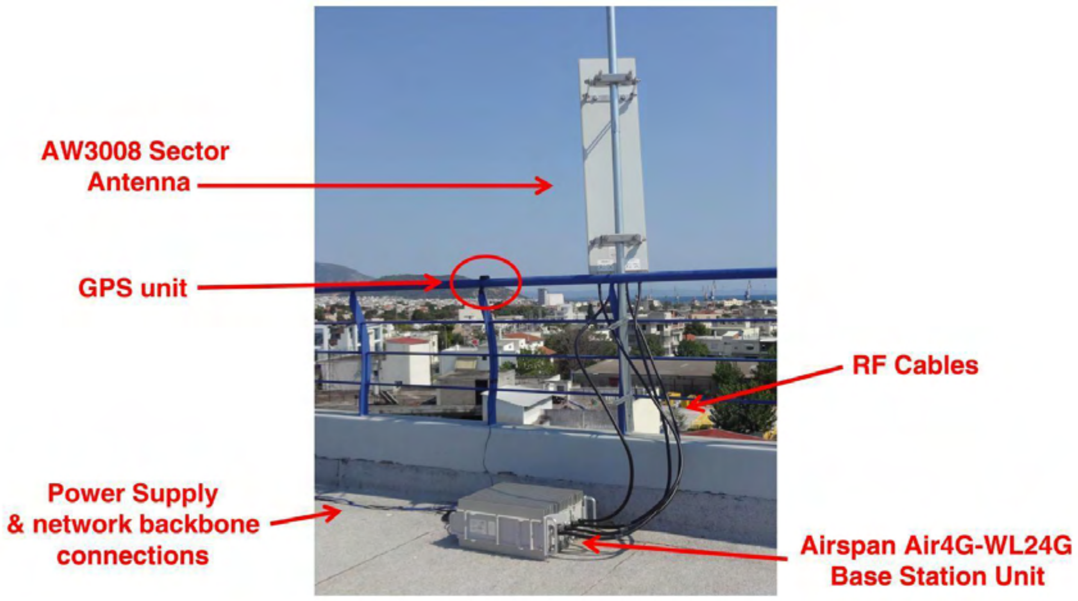

AW3008 Sector Antenna
GPS unit
RF Cables
Power Supply & network backbone connections
Airspan Air4G-WL24G Base Station Unit

Figure 9: WiMAX installation components

1.2.5 LTE testbed

LTE (Long Term Evolution), marketed as 4G LTE, is a standard for wireless communication of high-speed data for mobile phones and data terminals. It is based on the GSM/EDGE and UMTS/HSPA network technologies, increasing the capacity and speed using a different radio interface together with core network improvements. UTH in cooperation with COSMOTE, has acquired and configured several components. The core units are two LTE Access Points and the SIRRAN's LTEnet core network.

Figure 10: LTE access point

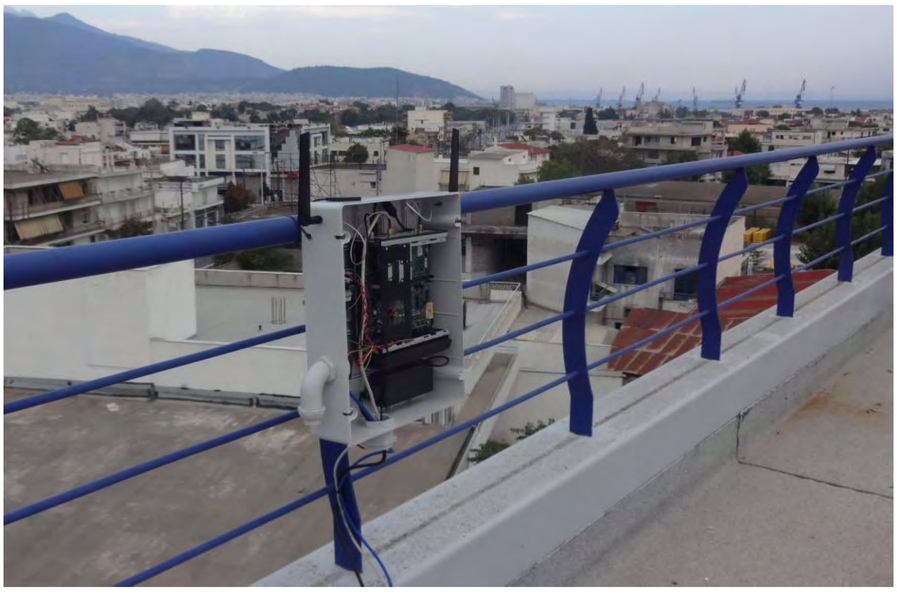


Figure 11: LTE installation in UTH premises

1.2.6 Facility architecture

The network architecture of the indoor and outdoor testbeds consists of three Gigabit Ethernet switches. Two Gigabit Ethernet switches interconnect the nodes with NITOS server of each testbed. The first one is the Control switch that provide for control of experiment execution and measurement collection and the second is the Experimental switch, which can be used for conducting wired experiments. A third Gigabit Ethernet, namely the Chassis Manager switch, is dedicated in controlling the operational status of the nodes through the transmission of custom http requests that control solid state relays on the Chassis Manager cards attached on each node.

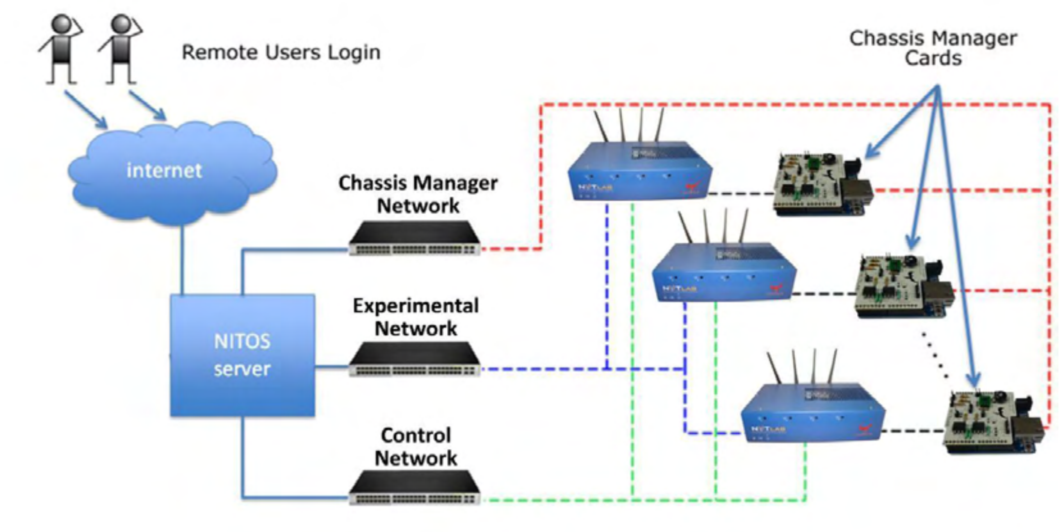


Figure 12: Outdoor/indoor testbeds network architecture

The overall architecture of the NITOS facility is demonstrated in Figure 13.

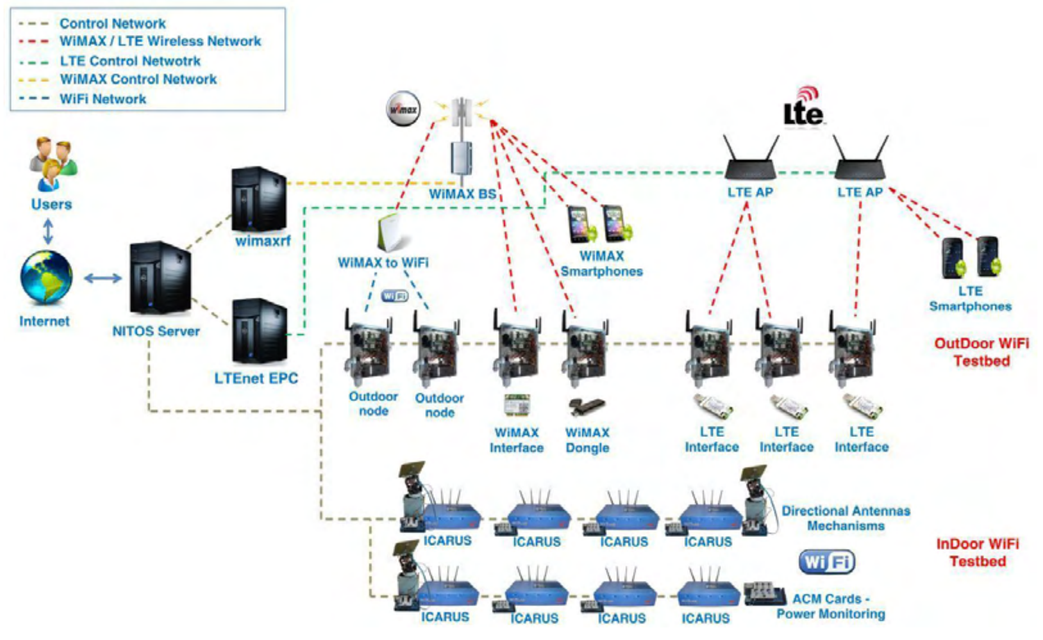


Figure 13: NITOS facility architecture

1.2.7 Broker

The broker can been seen as an aggregate manager for a testbed, which keeps the inventory with the available resources and provides a reference point where services related to experiment life cycle (resource discovery, reservation, provision) are concentrated.

The Broker is comprising a communication module with three different communication interfaces, an authentication and authorization module, a scheduler module, a database and an AM liaison module responsible for communicating with other OMF-6.0 RCs over FRCP.

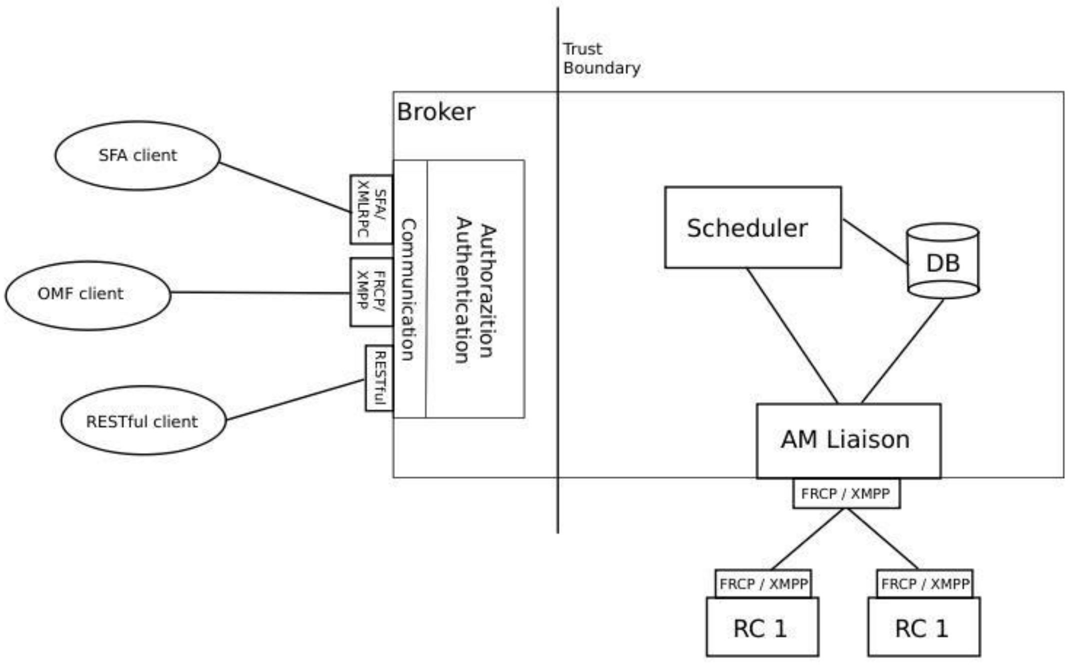


Figure 14: OMF Broker Architecture

The Authentication/Authorization module is adapted to the credentials that each of the APIs is using. Regarding Authentication, the process is based mainly on X.509 certificates that are used by the SFA authorities to authenticate users of different testbeds.

The module of scheduler, is responsible for scheduling the resources. When a user requests some resources for reserving them for a specific time, scheduler keeps all the necessary information in the corresponding database of the Broker. The database contains information regarding the availability of the resources based on the reservations done from the users and information regarding the description of the resources. The Broker was designed in order to be modular. The extension of the current inventory by adding new resources can easily achieved by defining a Class with the adequate properties that correspond to the properties of the new resources Also, the Scheduler allows the usage of different scheduling algorithms depending on the testbed owner's preferences.

The RESTful API provides interoperability with 3rd party tools or even testbed specific tools. Such an example is a web frontend of a testbed which enables their users to perform reservation of resources. The communication between the frontend and the backend which in this case is the Broker, can be easily made with the RESTful interface of the Broker.

Above we demonstrate an example of the FRPC messages during an operation of CMRC.

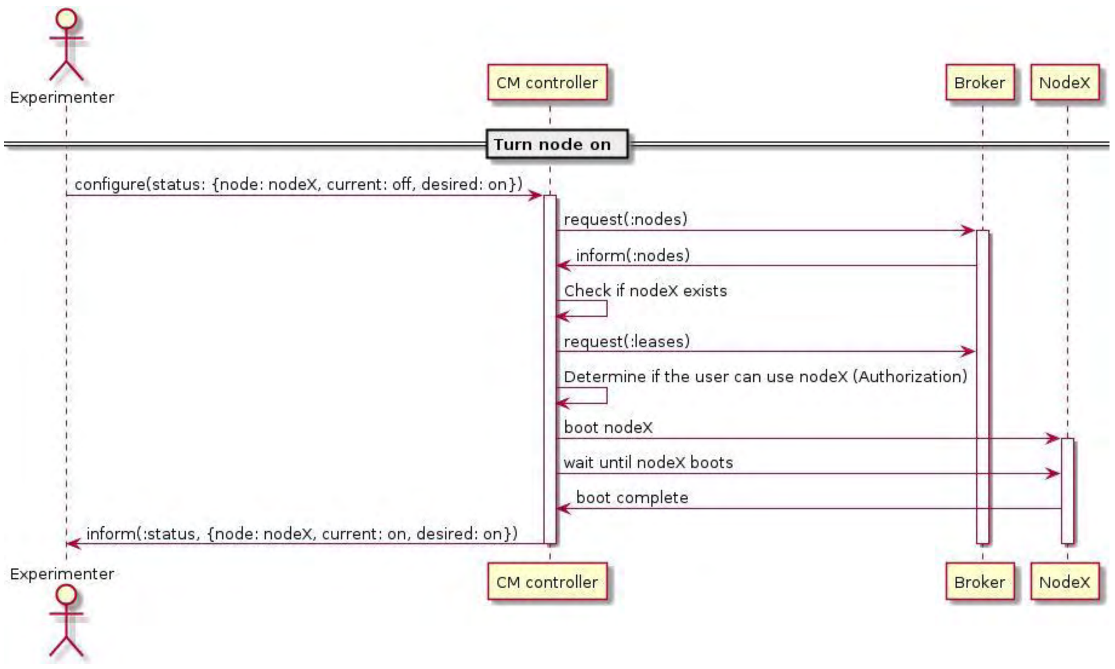
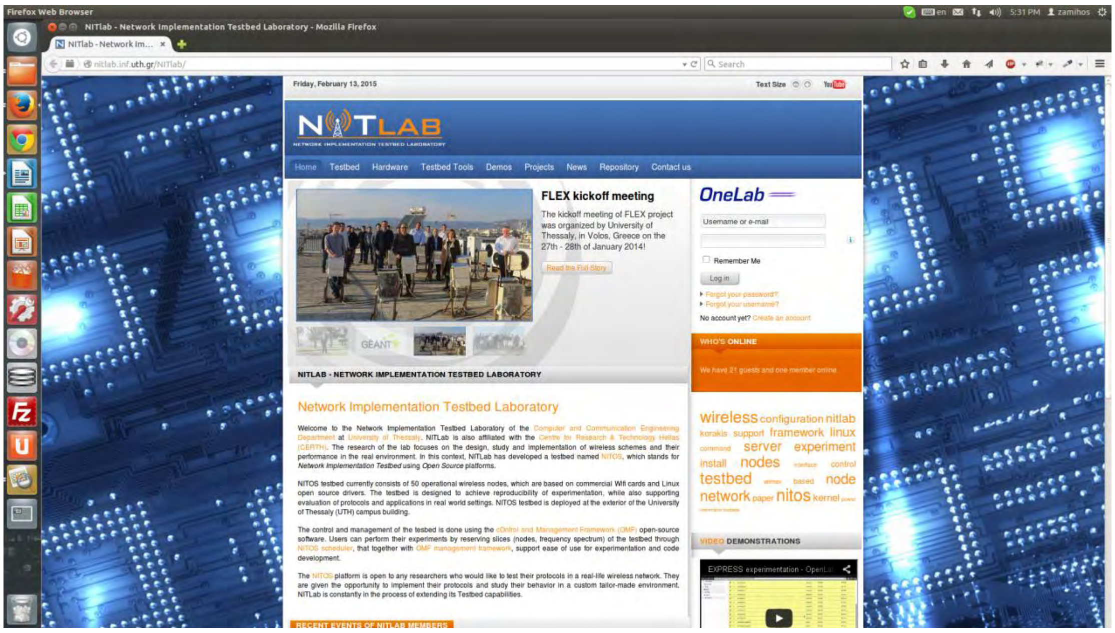


Figure 15: Interaction between Chassis Manager Card RC and Broker

1.3 Motivation

As we see, NITOS testbed provides a variety of technologies. Also, NITOS expands continuously with new technologies that require the corresponding interface in order to be configured by the users. This led us to develop a new Portal that incorporates all the experimentation tools of NITOS. Unlike the previous site of NITlab, our aim was to keep the information about the testbed and its facilities in NITlab's site and to separate the experimentation tools in a new Portal.


NITlab - Network Implementation Testbed Laboratory - Mozilla Firefox
nitlab.inf.uth.gr/NITlab/
Friday, February 13, 2015
Text Size
NITLAB
NETWORK IMPLEMENTATION TESTBED LABORATORY
Home
Testbed
Hardware
Testbed Tools
Demos
Projects
News
Repository
Contact us
OneLab
Username or e-mail
Remember Me
Log in
Forgot your password?
Forgot your username?
No account yet? Create an account
WHO'S ONLINE
VIDEO DEMONSTRATIONS
RECENT EVENTS OF NITLAB MEMBERS

FLEX kickoff meeting

The kickoff meeting of FLEX project was organized by University of Thessaly, in Volos, Greece on the 27th - 28th of January 2014!

Read the Full Story

We have 21 guests and one member online

NITLAB - NETWORK IMPLEMENTATION TESTBED LABORATORY

Network Implementation Testbed Laboratory

Welcome to the Network Implementation Testbed Laboratory of the Computer and Communication Engineering Department at University of Thessaly. NITLab is also affiliated with the Centre for Research & Technology Hellas (CERTH). The research of the lab focuses on the design, study and implementation of wireless schemes and their performance in the real environment. In this context, NITLab has developed a testbed named NITOS, which stands for *Network Implementation Testbed* using *Open Source* platforms.

NITOS testbed currently consists of 50 operational wireless nodes, which are based on commercial Wifi cards and Linux open source drivers. The testbed is designed to achieve reproducibility of experimentation, while also supporting evaluation of protocols and applications in real world settings. NITOS testbed is deployed at the exterior of the University of Thessaly (UTH) campus building.

The control and management of the tesbed is done using the cOntrol and Management Framework (OMF) open-source software. Users can perform their experiments by reserving slices (nodes, frequency spectrum) of the testbed through NITOS scheduler, that together with OMF management framework, support ease of use for experimentation and code development.

The NITOS platform is open to any researchers who would like to test their protocols in a real-life wireless network. They are given the opportunity to implement their protocols and study their behavior in a custom tailor-made environment. NITLab is constantly in the process of extending its Testbed capabilities.

Figure 16: NITlab's site Home Page

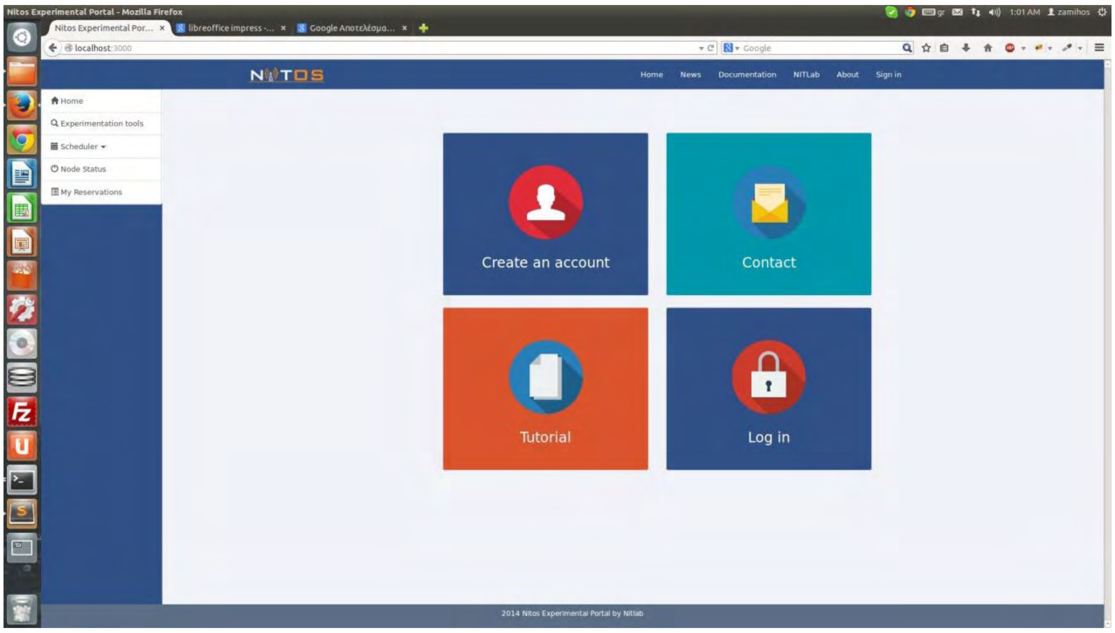


Figure 17: Home Page of NITOS Portal

2. Tools

In this section we will refer the tools that we use for the development of the Portal and the reasons that we choose them.

2.1 Ruby on Rails

The basic tool that we used was Ruby on Rails. Ruby on Rails is a web development framework written in the Ruby programming language. Since its debut in 2004, Ruby on Rails has rapidly became one of the most powerful and popular tools for building dynamic web applications.

Ruby on Rails is open-source and free for download and use. One reason that we chose Rails was its elegant and compact design. By exploiting the malleability of the underlying Ruby language, Rails effectively creates a domain-specific language for writing web applications. As a result, many common web programming tasks such as generating HTML, making data models, and routing URLs are easy with Rails, and the resulting application code is concise and readable. Rails also, provides a neat development environment to use MVC model.

A second reason was, that Ruby is the main development language used by the experimenters of NITlab.

Finally, Rails benefits from an unusually enthusiastic community. A rich variety of informative blogs and solutions to the most programming problems can be easily found. Also, a huge number of gems (self-contained solutions to specific problems such as calendar) can be used.

The version of Ruby that we use for the development was ruby 2.0.0p451 with Rails 4.1.1.

2.2 Structure and Style

For the development of the structure and the style of the Portal, HTML 5, CSS 3, Bootstrap and jQuery have been used.

HTML 5

Hyper Text Markup Language (HTML) is the standard markup language used to create web pages. For the construction of the Portal we used the fifth version of the HTML the HTML5.

CSS 3

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in HTML. We used CSS3 which is the latest version of CSS.

Bootstrap

Bootstrap is the most popular HTML, CSS and JavaScript framework for developing responsive, mobile-first web sites. This means the layout of the web pages adjusts dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone). Bootstrap also provides a number of re-usable components like buttons, buttons with drop-down option, navigation lists, horizontal and vertical tabs, navigation, pagination, etc.

jQuery

We also used jQuery which is the most popular JavaScript library in use today. jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML.

2.3 Gems

As we referred Ruby allows us to use gems as solution to specifics problems. Below we give a list of gems that we used for the development of the Portal.

- gem 'httparty'
- gem 'websocket-rails'
- gem 'momentjs-rails'
- gem 'bootstrap3-datetimepicker-rails'
- gem 'jquery-cookie-rails'

3. Development of the Portal

The development of the Portal can be separated into three main sections. The first section was the design of a database and a log in/out mechanism. The second section, was the development of four experimentation tools, namely, the Scheduler tools *Reservation* and *Quick Reservation*, *My Reservation* tool and *Node Status* tool. Finally, in the last section we dealt with the customization of the Portal.

In this chapter we discuss the development of the Portal. We separate this chapter in three subchapters. At first, we deal with the design and the construction of a database and a log in/out mechanism. Next, we present the architecture scheme of the Portal. Finally, we examine in detail each tool separately and give screen shots of the Portal.

3.1 Design Database and Log in/out Mechanism

The first thing that we dealt with, during the development of the Portal, was the design of a database and a log in/out mechanism. Our purpose was to give users the ability to sign up for our site and create a user profile page, so that they can use the experimentation tools.

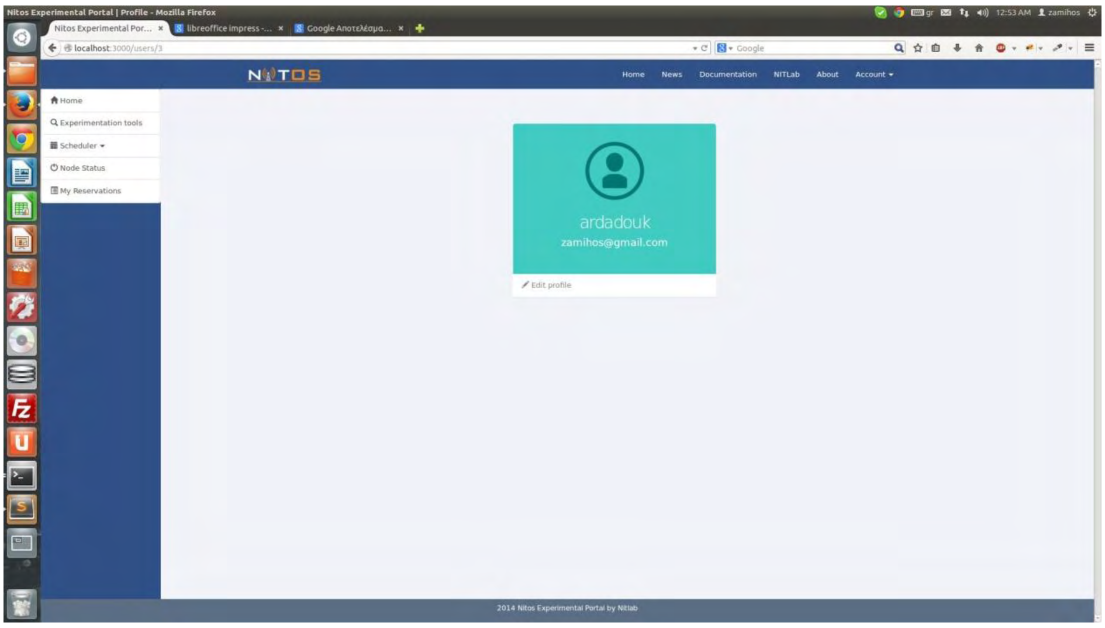
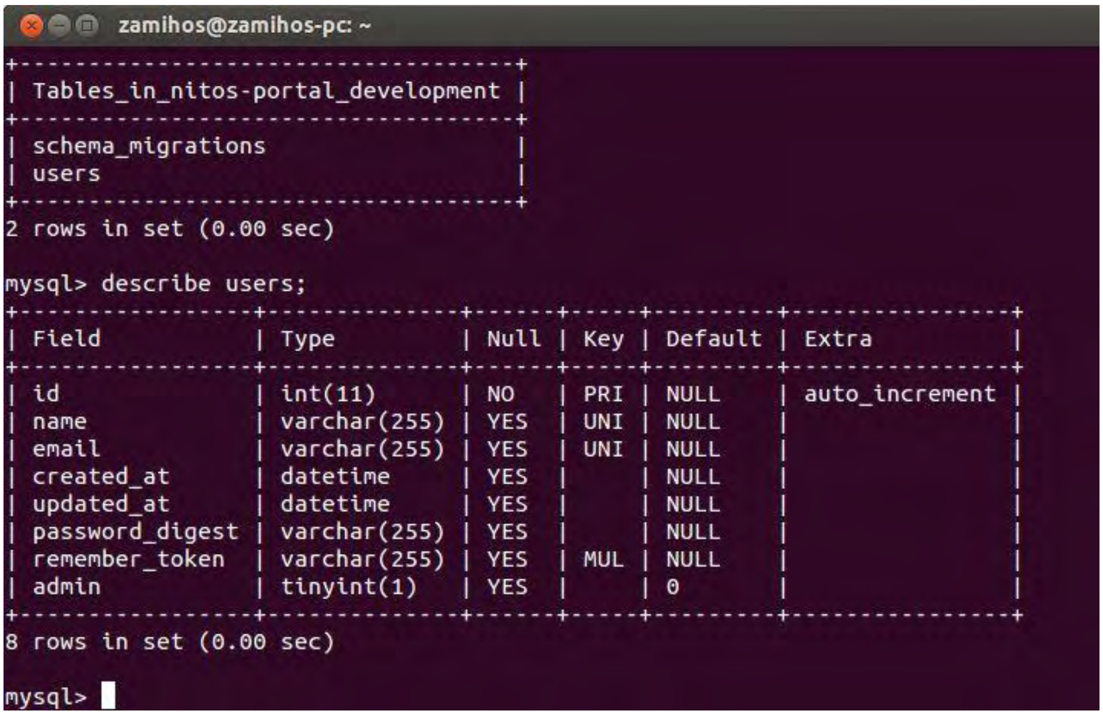


Figure 18: User's Profile Page

The columns of our database are:

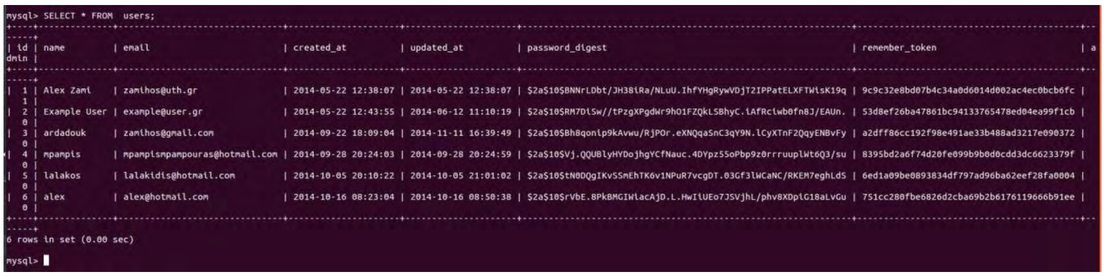
- id
- name
- email
- created_at
- updated_at
- password_digest
- remember_token



zamihos@zamihos-pc: ~

```
+-----------------------------------+
| Tables_in_nitos-portal_development |
+-----------------------------------+
| schema_migrations                 |
| users                             |
+-----------------------------------+
2 rows in set (0.00 sec)

mysql> describe users;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| id              | int(11)      | NO   | PRI | NULL    | auto_increment |
| name            | varchar(255) | YES  | UNI | NULL    |                |
| email           | varchar(255) | YES  | UNI | NULL    |                |
| created_at      | datetime     | YES  |     | NULL    |                |
| updated_at      | datetime     | YES  |     | NULL    |                |
| password_digest | varchar(255) | YES  |     | NULL    |                |
| remember_token  | varchar(255) | YES  | MUL | NULL    |                |
| admin           | tinyint(1)   | YES  |     | 0       |                |
+-----------------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)

mysql> 
```

Figure 19: Database scheme



```
mysql> SELECT * FROM  users;
+----+--------------+--------------------------------+---------------------+---------------------+--------------------------------------------------------------+------------------------------------------+--
-----+
| id | name         | email                          | created_at          | updated_at          | password_digest                                              | remember_token                           | a
dmin |
+----+--------------+--------------------------------+---------------------+---------------------+--------------------------------------------------------------+------------------------------------------+--
-----+
|  1 | Alex Zami    | zamihos@uth.gr                 | 2014-05-22 12:38:07 | 2014-05-22 12:38:07 | $2a$10$BNNrLDbt/JH38iRa/NLuU.IhfYHgRywVDjT2IPPatELXFTWisK19q | 9c9c32e8bd07b4c34a0d6014d002ac4ec0bcb6fc |
   1 |
|  2 | Example User | example@user.gr                | 2014-05-22 12:43:55 | 2014-06-12 11:10:19 | $2a$10$RM7DiSw//tPzgXPgdWr9hO1FZQkLSBhyC.iAfRciwb0fn8J/EAUn. | 53d8ef26ba47861bc94133765478ed04ea99f1cb |
   0 |
|  3 | ardadouk     | zamihos@gmail.com              | 2014-09-22 18:09:04 | 2014-11-11 16:39:49 | $2a$10$Bh8qonip9kAvwu/RjPOr.eXNQqaSnC3qY9N.lCyXTnF2QqyENBvFy | a2dff86cc192f98e491ae33b488ad3217e090372 |
   0 |
|  4 | mpampis      | mpampismpampouras@hotmail.com | 2014-09-28 20:24:03 | 2014-09-28 20:24:59 | $2a$10$Vj.QQUBlyHYDojhgYCfNauc.4DYpz55oPbp9z0rrruuplWt6Q3/su | 8395bd2a6f74d20fe099b9b0d0cdd3dc6623379f |
   0 |
|  5 | lalakos      | lalakidis@hotmail.com          | 2014-10-05 20:10:22 | 2014-10-05 21:01:02 | $2a$10$tN0DQgIKvSSmEhTK6v1NPuR7vcgDT.03Gf3lWCaNC/RKEM7eghLdS | 6ed1a09be0893834df797ad96ba62eef28fa0004 |
   0 |
|  6 | alex         | alex@hotmail.com               | 2014-10-16 08:23:04 | 2014-10-16 08:50:38 | $2a$10$rVbE.8PkBMGIWiacAjD.L.HwIiUEo7JSVjhL/phv8XDpiG18aLvGu | 751cc280fbe6826d2cba69b2b6176119666b91ee |
   0 |
+----+--------------+--------------------------------+---------------------+---------------------+--------------------------------------------------------------+------------------------------------------+--
-----+
6 rows in set (0.00 sec)

mysql> 
```

Figure 20: Data in Database

Every user that wants to have access to Portal's tools, has to complete a form with his/her user name, email and password.

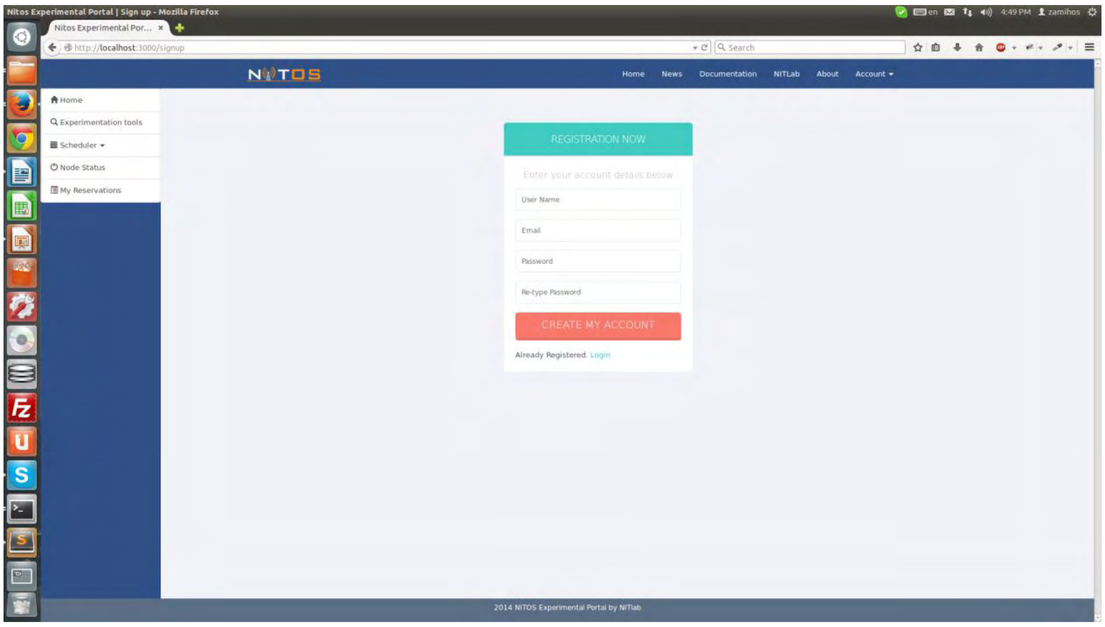
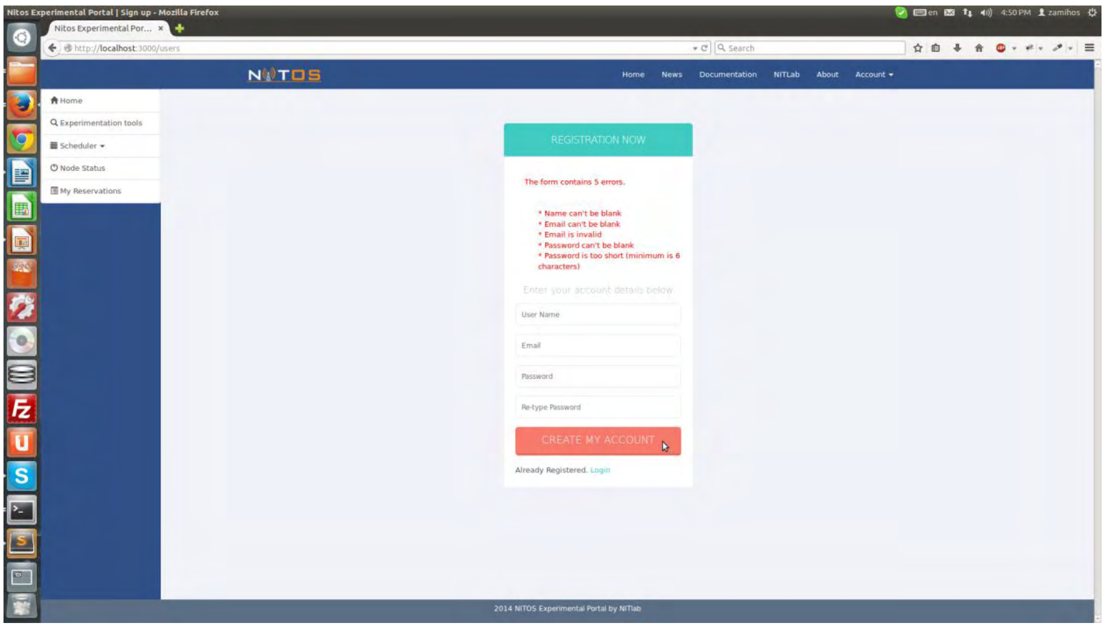

Nitos Experimental Portal | Sign up - Mozilla Firefox
http://localhost:3000/signup
NITOS
Home
News
Documentation
NITLab
About
Account
Home
Experimentation tools
Scheduler
Node Status
My Reservations
REGISTRATION NOW
Enter your account details below
User Name
Email
Password
Re-type Password
CREATE MY ACCOUNT
Already Registered. Login
2014 NITOS Experimental Portal by NITlab

Figure 21: Registration Form

We added some validations about the user's data. *User's name* should not be blank, it has to be less than 50 characters and to be unique. *Email* should not be blank, it has to be unique and should match the specific format characteristics of email addresses.

Password should not be blank and it has to be at least 6 characters.



The form contains 5 errors.

* Name can't be blank
* Email can't be blank
* Email is invalid
* Password can't be blank
* Password is too short (minimum is 6 characters)

Enter your account details below

Already Registered. Login

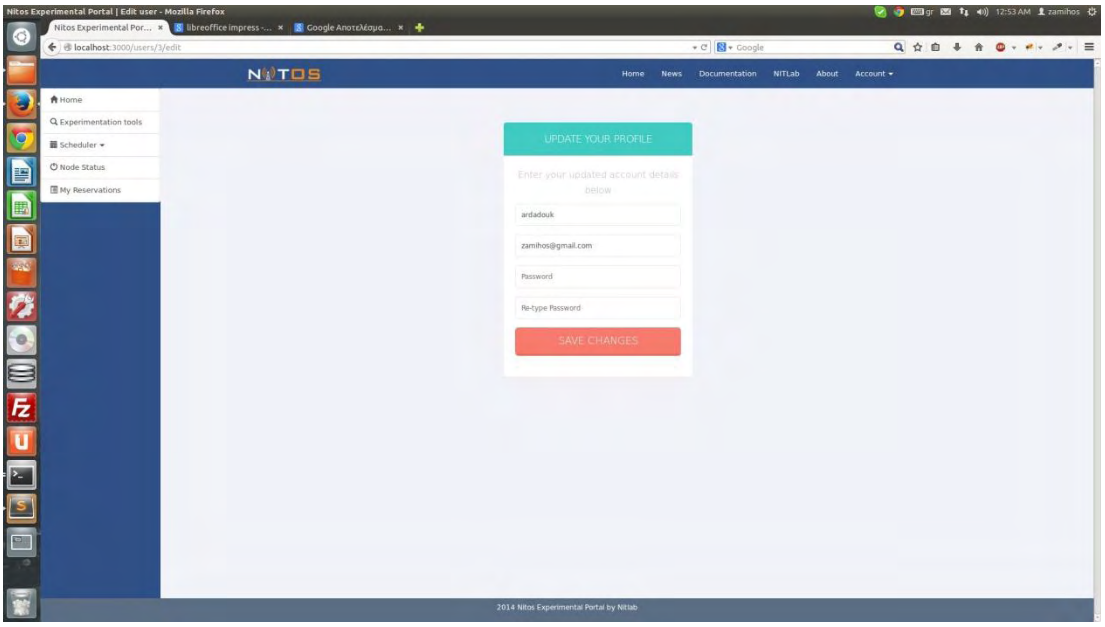
Figure 22: Validation check

We added *password_digest* column to the users table to store a hashed version of the user's password in the database and use it for authentication. The method for authenticating users is to take a submitted password, hash it, and compare the result to the hashed value stored in the database. If the two match, then the submitted password is correct and the user is authenticated. By comparing hashed values instead of raw passwords, we are able to authenticate users without storing the passwords themselves. This means that, even if our database is compromised, our users' passwords are still secure.

We also added *remember_token,* a unique, secure remember token for storing it as a permanent cookie. We used urlsafe_base64 method from the SecureRandom module in the Ruby standard library that returns a random string of length 16 composed of the characters A–Z, a–z, 0–9, "-", and "_". We store the base64 token on browser, and then store its hash digest in the database. We can then sign users in automatically by retrieving the token from the cookie, calculating the hash digest, and then searching for a remember token matching the digest's value. The reason for storing only hashed tokens is so that, even if our entire database is compromised, the attacker still won't be able to use the remember tokens to sign in. To make our remember token even more secure, we change it every time a user creates a new session, which means that any hijacked sessions will expire the next time a user signs in.

Id column created automatically, and used by Rails to identify each row of the database uniquely. *Created_at* and *updated_at* are timestamps that automatically record when a given user is created and updated. Finally, we added the admin attribute, to identify administrative users of the site.

We also give user the opportunity to update his/her data.


Nitos Experimental Portal | Edit user - Mozilla Firefox
localhost:3000/users/3/edit
NITOS
Home
News
Documentation
NITLab
About
Account
Home
Experimentation tools
Scheduler
Node Status
My Reservations
2014 Nitos Experimental Portal by Nitlab

UPDATE YOUR PROFILE

Enter your updated account details below

ardadouk

zamihos@gmail.com

Password

Re-type Password

SAVE CHANGES

Figure 23: Updating Profile Page

3.2 NITOS Portal Architecture

The figure below shows the architecture of NITOS Portal.

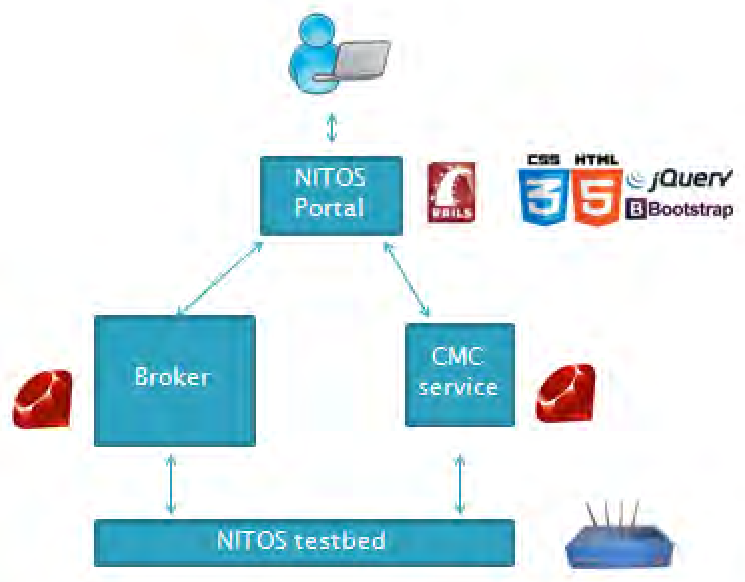


Figure 24: NITOS Portal Architecture

NITOS Portal gives user the opportunity to use its experimentation tools. All the tools communicates with Broker to discover the reservations of the users. However, we can categorize these tools in two groups.

The first group consists of the Scheduler tools (Reservation, Quick Reservation) and My Reservation tool, which communicate with Broker via http requests. Broker is responsible for receiving the incoming commands, parsing the commands, making the appropriate actions - like make or cancel a reservation- and responding back to the Portal.

The second group consists of the Node Status tool, which communicates with the Chassis Manager (CM) cards. The CM cards runs a tiny web-server which is responsible for receiving incoming commands via http requests, parsing the commands, making the appropriate actions - like turn a node on/off or reset a node - and responding back with a standard http response.

3.3 Experimentation Tools

In this section we discuss detailed the development of the experimentation tools. We display some screen shots of the new web interface and we compare it with the previous version of the tools.

To use the experimentation tool user requires to have an account on Broker with the same user name as on Portal. Furthermore, a user should has at least one slice for being able to experimenting. To create slices and an account on Broker, user should has to contact with NITlab's team.

3.3.1 User's Timezone Detection

All the experimentation tools display every information according to user's timezone. To achieve that we have to detect correctly the user's timezone. We chose that, from explicitly ask them to tell their timezones, because if a user travels, he/her will has to change his/her timezone setting again. Since computer already has a timezone setting, and the browser (via Javascript) already knows what it is, we should take that info pass it to the server when a request is made.

The Javascript Date object has a getTimezoneOffset method which returns an integer timezone offset, but daylight savings time complicates things. We also wanted the timezone represented as a string to hand to Rails, like “Europe/Athens”, not an integer offset. For that purpose we used jsTimezoneDetect library. This let us get the timezone like this:

```
  var tz = jstz.determine();

  tz.name(); // returns “Europe/Helsinki”
```

The next step was to send that information to server. For that purpose we use “jquery-cookie-rails” gem, for sending the user's current timezone via a cookie. Then in Rails app, we use an arround_filter in ApplicationController to set the timezone for one request only, and reset it after the request is done. The whole code that we use:

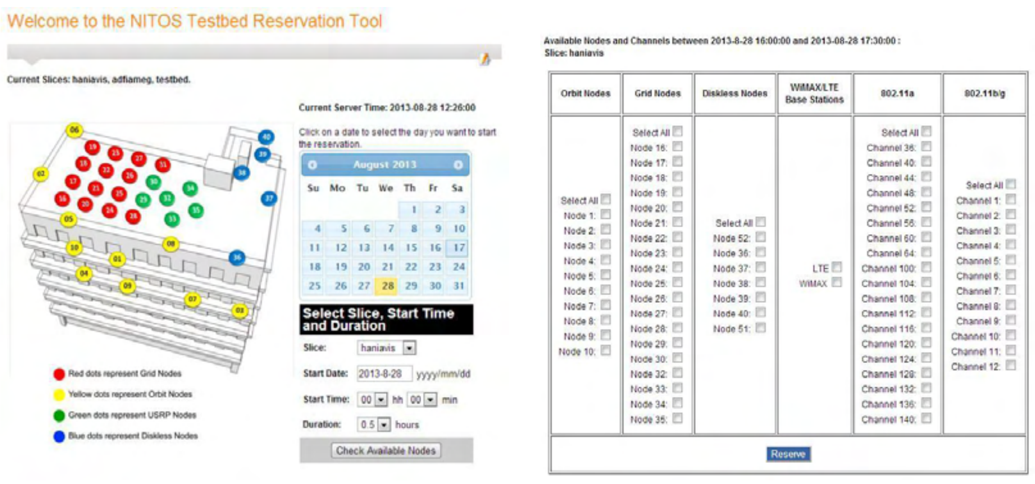
```
<script>

        jQuery(function() {

        var tz = jstz.determine();

        $.cookie('timezone', tz.name(), { path: '/' });

        });

</script>


def set_timezone

   default_timezone = Time.zone

   client_timezone  = cookies[:timezone]

   Time.zone = client_timezone if client_timezone.present?

   yield

  ensure

   Time.zone = default_timezone

  end

end
```

If something goes wrong we set the default timezone as the user's timezone. We set the default timezone to Athens timezone (GMT +2).

As Broker stores all the information about the reservations of the resources in UTC, every time that we make a GET request to Broker we convert the times to user's timezone.

3.3.2 Scheduler Tools

Experimenting on NITOS depends on using its resources (nodes, channels). For that purpose NITlab's team had created NITOS Scheduler, a tool which is responsible for managing the testbed's resources. NITOS Scheduler was mainly consisting of two components, a web-based user interface and a server component compromising of some scripts and a database.



Welcome to the NITOS Testbed Reservation Tool
Red dots represent Grid Nodes
Yellow dots represent Orbit Nodes
Green dots represent USRP Nodes
Blue dots represent Diskless Nodes
August 2013
Select Slice, Start Time and Duration
Check Available Nodes
Reserve

Current Slices: haniavis, adfiameg, testbed.

Current Server Time: 2013-08-28 12:26:00

Click on a date to select the day you want to start the reservation.

Slice: haniavis

Start Date: 2013-8-28 yyyy/mm/dd

Start Time: 00 hh 00 min

Duration: 0.5 hours

Available Nodes and Channels between 2013-8-28 16:00:00 and 2013-08-28 17:30:00 :
Slice: haniavis

Orbit Nodes	Grid Nodes	Diskless Nodes	WiMAX/LTE Base Stations	802.11a	802.11b/g
Select All ☐ Node 1: ☐ Node 2: ☐ Node 3: ☐ Node 4: ☐ Node 5: ☐ Node 6: ☐ Node 7: ☐ Node 8: ☐ Node 9: ☐ Node 10: ☐	Select All ☐ Node 16: ☐ Node 17: ☐ Node 18: ☐ Node 19: ☐ Node 20: ☐ Node 21: ☐ Node 22: ☐ Node 23: ☐ Node 24: ☐ Node 25: ☐ Node 26: ☐ Node 27: ☐ Node 28: ☐ Node 29: ☐ Node 30: ☐ Node 32: ☐ Node 33: ☐ Node 34: ☐ Node 35: ☐	Select All ☐ Node 52: ☐ Node 36: ☐ Node 37: ☐ Node 38: ☐ Node 39: ☐ Node 40: ☐ Node 51: ☐	LTE ☐ WiMAX ☐	Select All ☐ Channel 36: ☐ Channel 40: ☐ Channel 44: ☐ Channel 48: ☐ Channel 52: ☐ Channel 56: ☐ Channel 60: ☐ Channel 64: ☐ Channel 100: ☐ Channel 104: ☐ Channel 108: ☐ Channel 112: ☐ Channel 116: ☐ Channel 120: ☐ Channel 124: ☐ Channel 128: ☐ Channel 132: ☐ Channel 136: ☐ Channel 140: ☐	Select All ☐ Channel 1: ☐ Channel 2: ☐ Channel 3: ☐ Channel 4: ☐ Channel 5: ☐ Channel 6: ☐ Channel 7: ☐ Channel 8: ☐ Channel 9: ☐ Channel 10: ☐ Channel 11: ☐ Channel 12: ☐

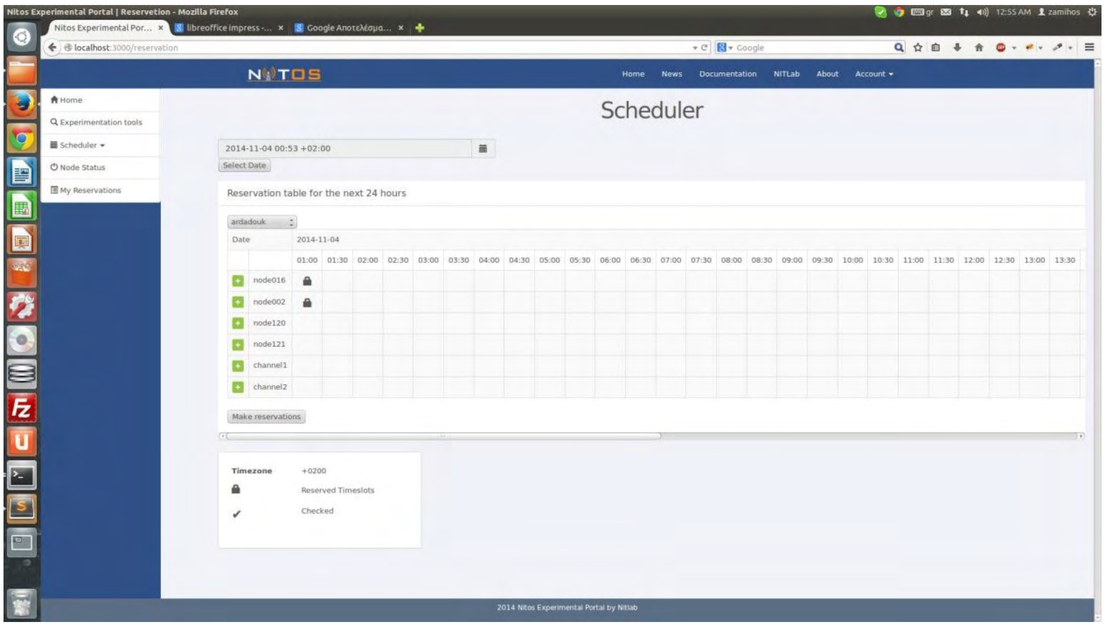
Figure 25: Previous NITOS Scheduler's web interface

In this thesis we tried to develop a new way for managing the testbed resources. We developed a new web-based interface, consisting of Scheduler's tools (Reservation and Quick Reservation), which communicates with Broker. Then, the Broker's module of scheduler, is responsible for scheduling the resources and for keeping all the necessary information in the corresponding database of the Broker. Since NITOS objective is to serve as many users as possible without any complicated procedures, our scheduler is developed in that spirit.

3.3.2.1 Reservation Tool

Reservation tool is the main tool to discover the available resources (nodes, channels). The experimenter also has the opportunity to compare the technical characteristics of the resources and reserve some of them.

By accessing the Reservation tool, user has the opportunity to discover the available resources for the next 24 hours (in time slots of 30 min), according to the information stored in the database of Broker. All the information is displayed according to user's timezone. We chose to display that information in a two dimensional table with the names of the resources as rows and the time slots as columns. Every cell of that table represents the availability of a resource in a specific time slot. If a resource is reserved, a lock is shown in the corresponding cell, otherwise the cell is empty.


Scheduler

2014-11-04 00:53 +02:00

Select Date

Reservation table for the next 24 hours

ardadouk

Date	2014-11-04																									
	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30	07:00	07:30	08:00	08:30	09:00	09:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30
node016	🔒																									
node002	🔒																									
node120																										
node121																										
channel1																										
channel2																										

Make reservations

Timezone	+0200
🔒	Reserved Timeslots
✓	Checked

Nitos Experimental Portal | Reservation - Mozilla Firefox
localhost:3000/reservation
Home
News
Documentation
NITLab
About
Account
Home
Experimentation tools
Scheduler
Node Status
My Reservations
2014 Nitos Experimental Portal by Nitlab

Figure 26: Reservation Tool – First Page

We also give the opportunity to the experimenter to choose a specific date in the future and discover the schedule of the resources for that day.

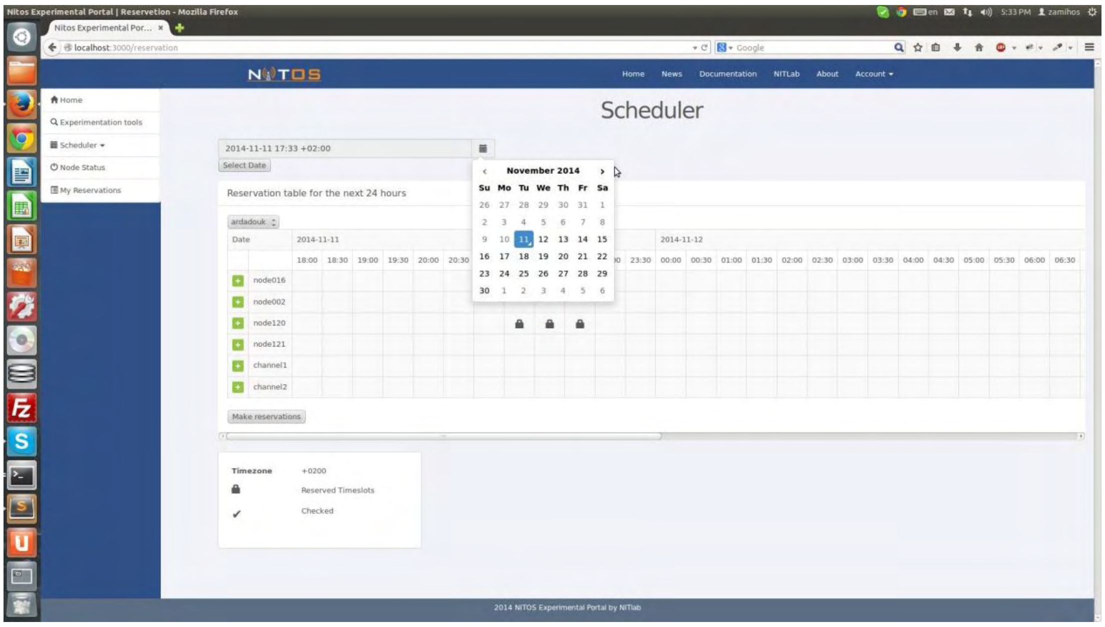


Figure 27: Reservation Tool - Choose Date

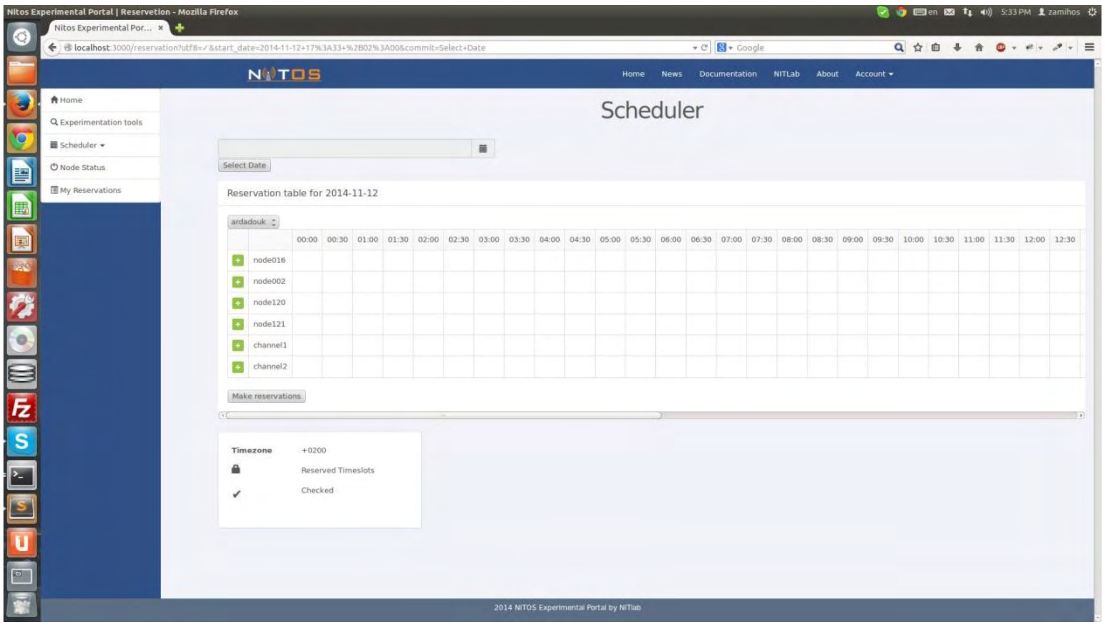
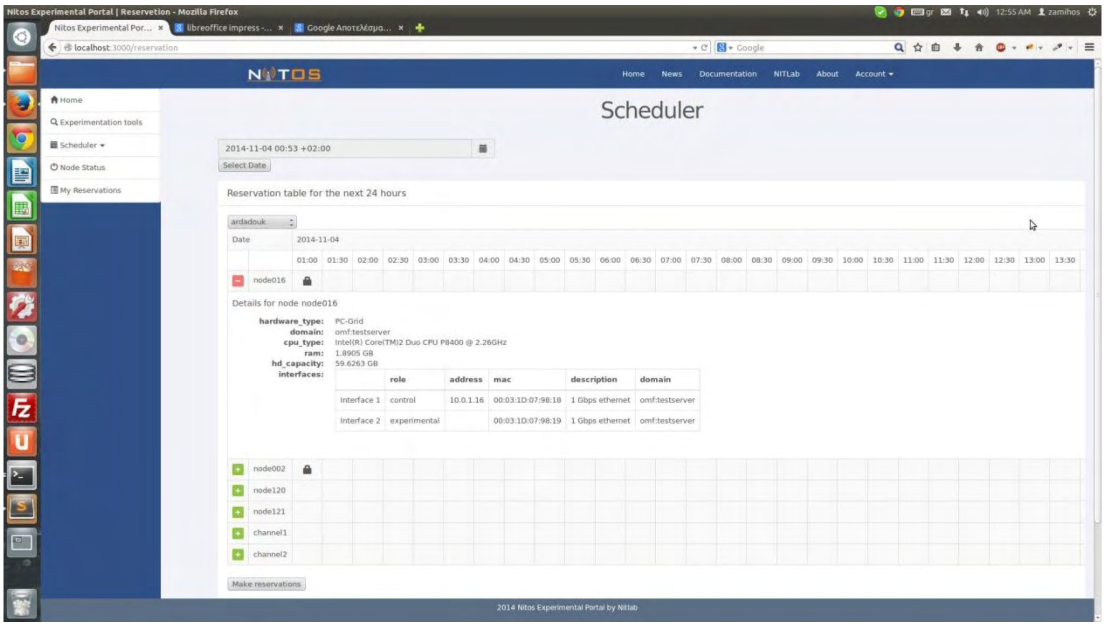


Figure 28: Reservation Tool – Schedule of chosen date

Reservation tool also displays information for the technical characteristics of every resource. So, the users can compare the resources and select the appropriate resources for experimenting. The characteristics that we chose to display for nodes are hardware type, domain, cpu type, ram, hd capacity and the interfaces, while for channels we chose frequency and domain.



Nitos Experimental Portal | Reservation - Mozilla Firefox
localhost:3000/reservation
NITOS
Home
News
Documentation
NITLab
About
Account
Home
Experimentation tools
Scheduler
Node Status
My Reservations
Select Date
2014 Nitos Experimental Portal by Nitlab

Scheduler

2014-11-04 00:53 +02:00

Reservation table for the next 24 hours

ardadouk

Date		2014-11-04																									
		01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30	07:00	07:30	08:00	08:30	09:00	09:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30
−	node016	🔒																									

Details for node node016

hardware_type:	PC-Grid
domain:	omf:testserver
cpu_type:	Intel(R) Core(TM)2 Duo CPU P8400 @ 2.26GHz
ram:	1.8905 GB
hd_capacity:	59.6263 GB
interfaces:	

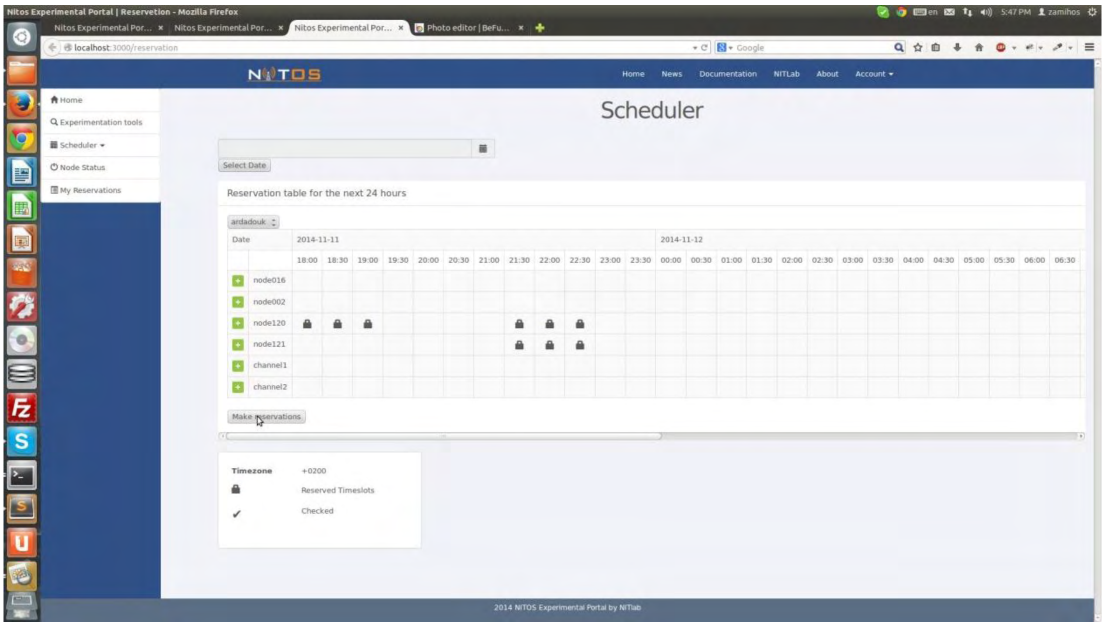
	role	address	mac	description	domain
Interface 1	control	10.0.1.16	00:03:1D:07:98:18	1 Gbps ethernet	omf:testserver
Interface 2	experimental		00:03:1D:07:98:19	1 Gbps ethernet	omf:testserver

		01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30	07:00	07:30	08:00	08:30	09:00	09:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30
+	node002	🔒																									
+	node120																										
+	node121																										
+	channel1																										
+	channel2																										

Make reservations

Figure 29: Reservation Tool – Resource's technical characteristics

Except for discovering the availability of the resources, Reservation tool gives users the opportunity to make a reservation on the available resources. User may select the slice which prefers for reserving the resources. Making a reservation requires from user to check more than one cell of the table. If no resources are checked and user press the "Make reservations" button a relative message will be shown. A message will inform the user that he/her should check at least a cell of the table for making a reservation.



Nitos Experimental Portal | Reservation - Mozilla Firefox
localhost:3000/reservation
NITOS
Home
News
Documentation
NITLab
About
Account
Home
Experimentation tools
Scheduler
Node Status
My Reservations
Select Date
2014 NITOS Experimental Portal by NITlab

Scheduler

Reservation table for the next 24 hours

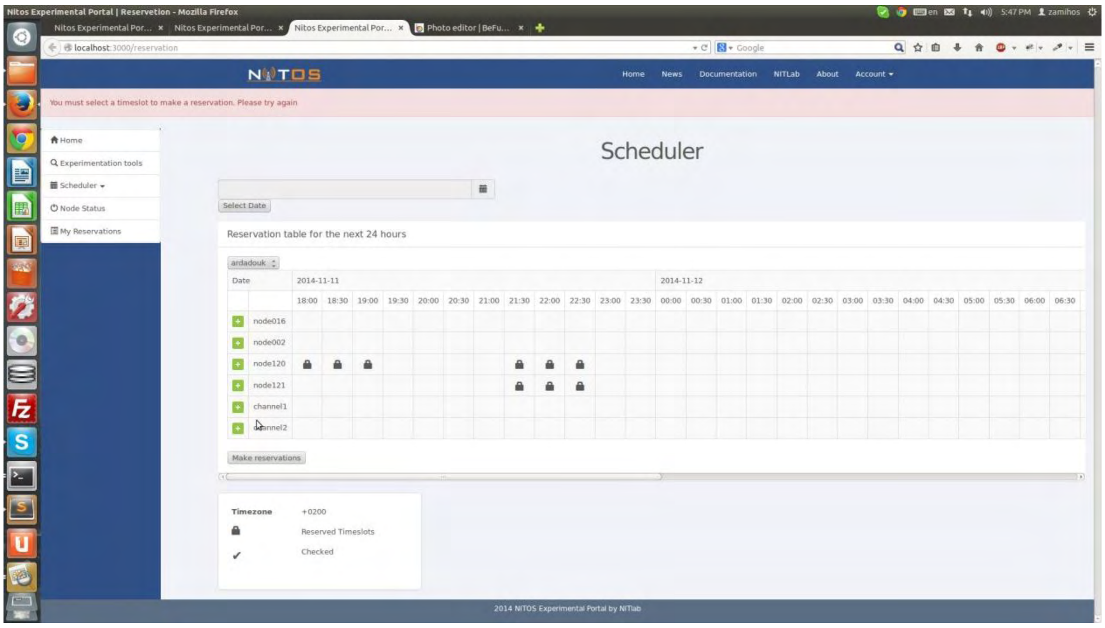
ardadouk

Date		2014-11-11												2014-11-12													
		18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30
+	node016																										
+	node002																										
+	node120	🔒	🔒	🔒					🔒	🔒	🔒																
+	node121								🔒	🔒	🔒																
+	channel1																										
+	channel2																										

Make reservations

Timezone	+0200
🔒	Reserved Timeslots
✓	Checked

Figure 30: Reservation Tool – Empty Reservation



You must select a timeslot to make a reservation. Please try again

Scheduler

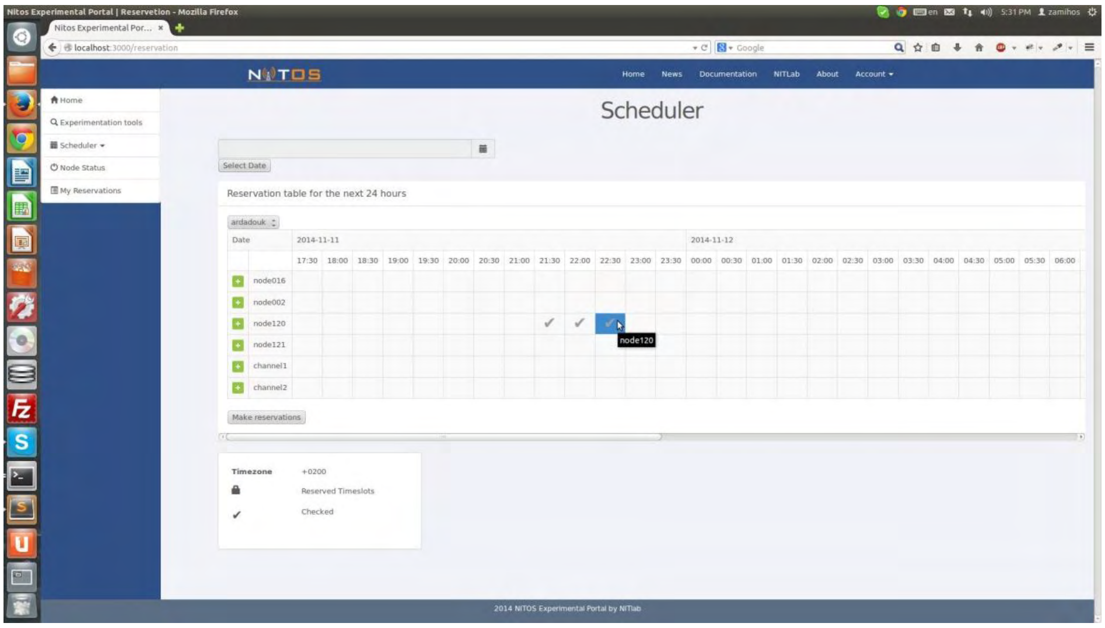
Reservation table for the next 24 hours

Date	2014-11-11												2014-11-12													
	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30
node016																										
node002																										
node120	🔒	🔒	🔒					🔒	🔒	🔒																
node121								🔒	🔒	🔒																
channel1																										
channel2																										

Timezone	+0200
🔒	Reserved Timeslots
✓	Checked

Figure 31: Reservation Tool – Empty Reservation Failure Message

If a cell is checked, a tick symbol is shown in the corresponding cell.



Scheduler

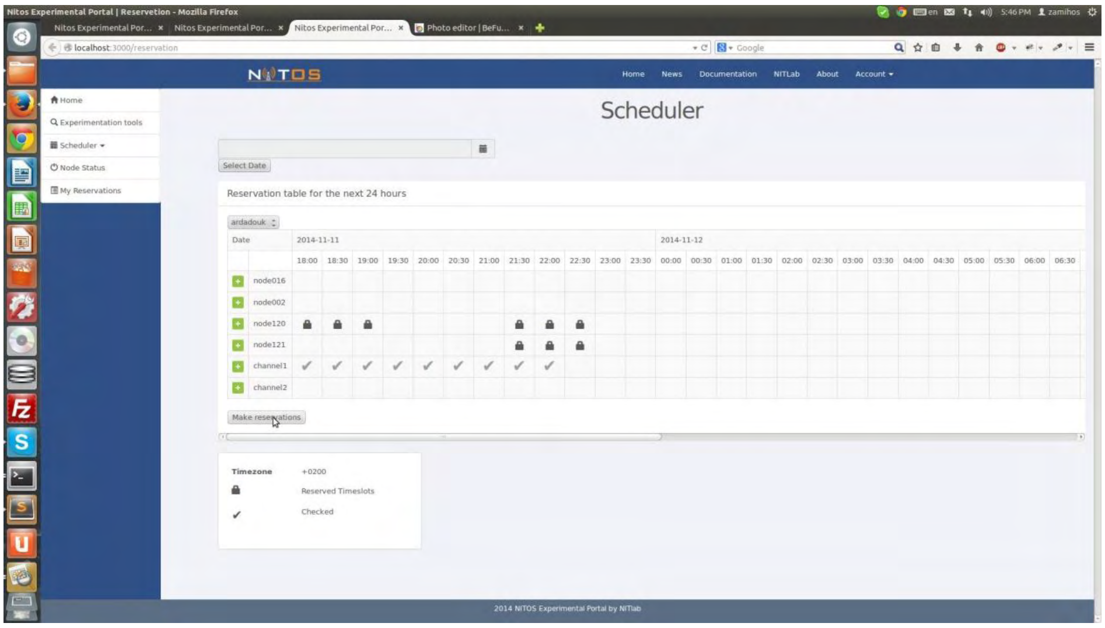
Reservation table for the next 24 hours

Date	2014-11-11													2014-11-12												
	17:30	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00
node016																										
node002																										
node120									✓	✓	✓															
node121																										
channel1																										
channel2																										

Timezone	+0200
🔒	Reserved Timeslots
✓	Checked

Figure 32: Reservation Tool – Choose time slots

On every reservation there is a limit about the number of time slots which users can reserve for every resource. One user can reserve until 8 time slots (4 hours) of every resource, per reservation. If more than 8 time slots are chosen a relative message will be shown.



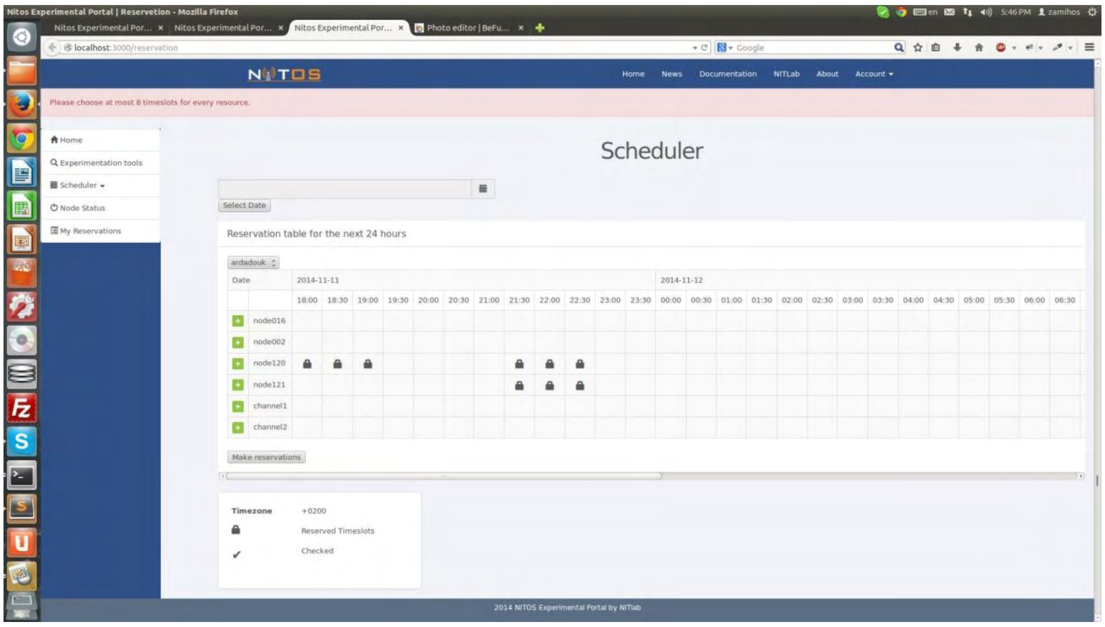
Scheduler

Reservation table for the next 24 hours

Date	2014-11-11												2014-11-12													
	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30
node016																										
node002																										
node120	🔒	🔒	🔒					🔒	🔒	🔒																
node121								🔒	🔒	🔒																
channel1	✓	✓	✓	✓	✓	✓	✓	✓	✓																	
channel2																										

Timezone		+0200
🔒		Reserved Timeslots
✓		Checked

Figure 33: Reservation Tool – Trying to reserve more than 8 time slots



Please choose at most 8 timeslots for every resource.

Scheduler

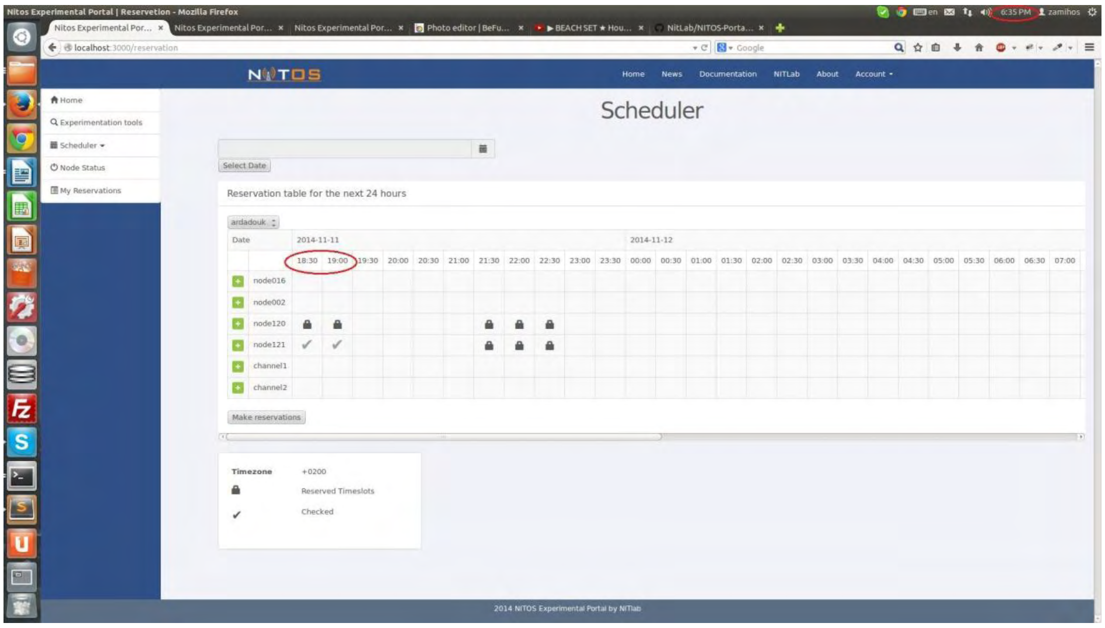
Reservation table for the next 24 hours

Date	2014-11-11												2014-11-12													
	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30
node016																										
node002																										
node120	🔒	🔒	🔒					🔒	🔒	🔒																
node121								🔒	🔒	🔒																
channel1																										
channel2																										

Timezone		+0200
🔒		Reserved Timeslots
✓		Checked

Figure 34: Reservation Tool – Limitation Failure Message

Furthermore, if user selects a "forgotten time slot", we alert him and urging him to try again, on the updated schedule. We use the term of the forgotten time slot to describe the case when an experimenter tries to make a reservation on a time slot that has gone.



Scheduler

Select Date

Reservation table for the next 24 hours

ardadouk

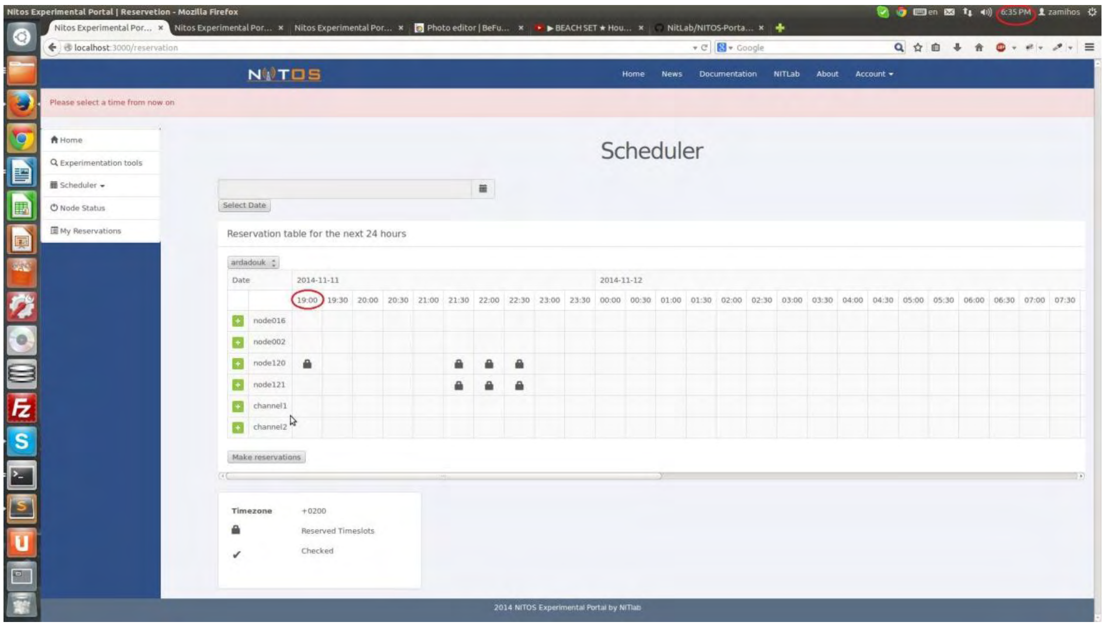
Date	2014-11-11											2014-11-12														
	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30	07:00
node016																										
node002																										
node120	🔒	🔒					🔒	🔒	🔒																	
node121	✓	✓					🔒	🔒	🔒																	
channel1																										
channel2																										

Make reservations

Timezone	+0200
🔒	Reserved Timeslots
✓	Checked

2014 NITOS Experimental Portal by NITlab

Figure 35: Reservation Tool – Forgotten time slots case



Please select a time from now on

Scheduler

Select Date

Reservation table for the next 24 hours

ardadouk

Date	2014-11-11										2014-11-12															
	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30	07:00	07:30
node016																										
node002																										
node120	🔒					🔒	🔒	🔒																		
node121						🔒	🔒	🔒																		
channel1																										
channel2																										

Make reservations

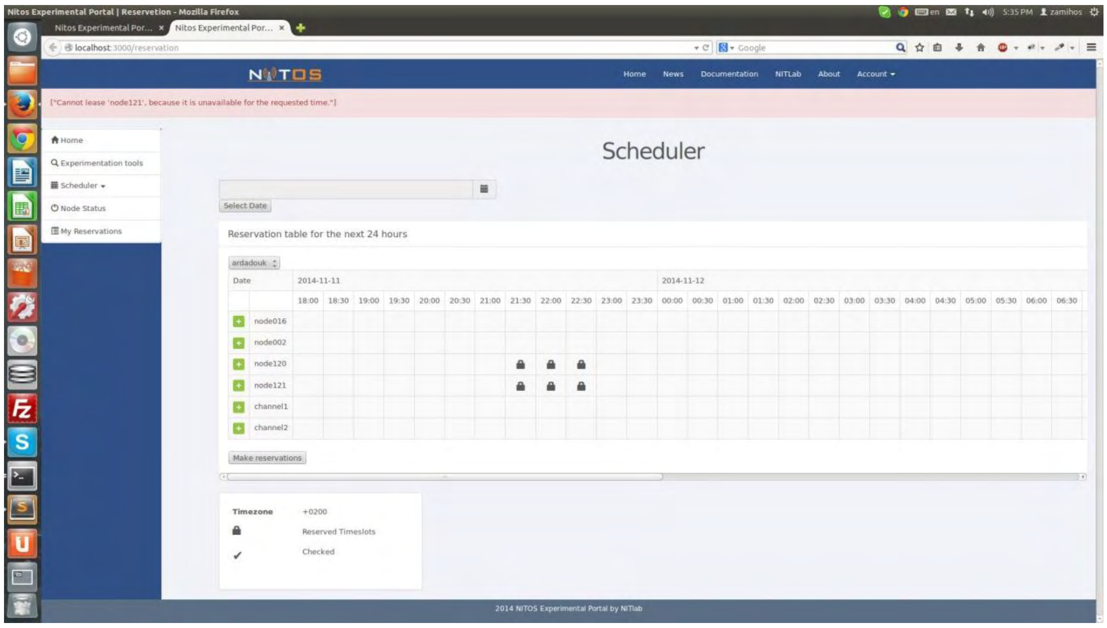
Timezone	+0200
🔒	Reserved Timeslots
✓	Checked

2014 NITOS Experimental Portal by NITlab

Figure 36: Reservation Tool – Forgotten time slot updated schedule

The last case which a reservation may fail is when another user reserve the same resource before. As the Portal can be used by many experimenters simultaneously, they may try to reserve a node that has been already reserved by another user. In that case Portal will display a relative message and the refreshed table.

As a case scenario assume that user A opens the Reservation tool. He/her checks some time slots of node121 for making a reservation. Simultaneously a second user (user B) accesses the reservation tool and selects the same or some of the time slots that the user A tries to reserve. User B presses the "Make reservation" button first and makes a successful reservation for the node121. When, finally, user A presses the "Make reservation" button a message will be shown which will inform user A that this resource is unavailable for the selected time slots and the updated schedule will be displayed to the user.


Nitos Experimental Portal | Reservetion - Mozilla Firefox

localhost:3000/reservation

["Cannot lease 'node121', because it is unavailable for the requested time."]

Scheduler

Reservation table for the next 24 hours

ardadouk

Date	2014-11-11												2014-11-12													
	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30
node016																										
node002																										
node120								🔒	🔒	🔒																
node121								🔒	🔒	🔒																
channel1																										
channel2																										

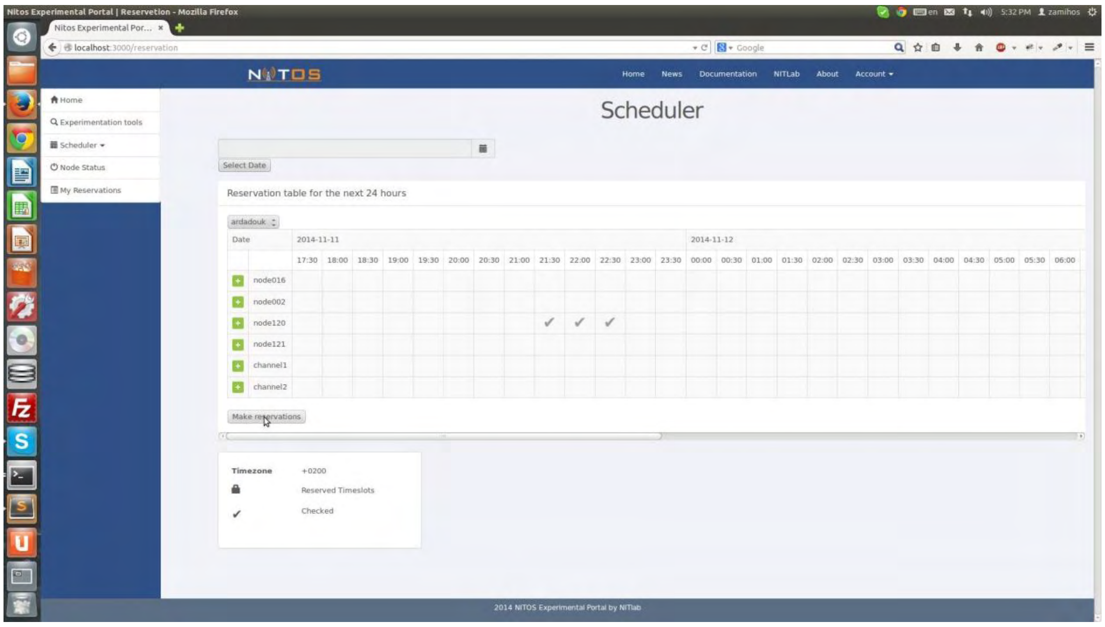
Make reservations

Timezone	+0200
🔒	Reserved Timeslots
✓	Checked

2014 NITOS Experimental Portal by NITlab

Figure 37: Reservation Tool – Failure because of other user

If none of the above cases happen, a success message will be displayed.


Nitos Experimental Portal | Reservetion - Mozilla Firefox

localhost:3000/reservation

Scheduler

Reservation table for the next 24 hours

ardadouk

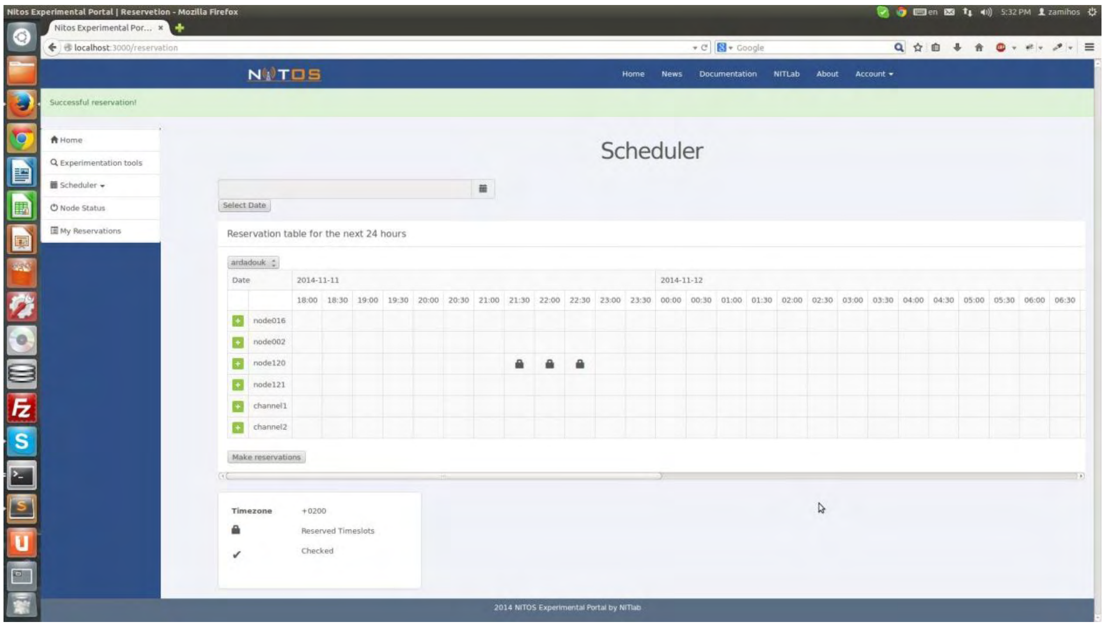
Date	2014-11-11													2014-11-12												
	17:30	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00
node016																										
node002																										
node120									✓	✓	✓															
node121																										
channel1																										
channel2																										

Make reservations

Timezone	+0200
🔒	Reserved Timeslots
✓	Checked

2014 NITOS Experimental Portal by NITlab

Figure 38: Reservation Tool – Make Reservation


Nitos Experimental Portal | Reservetion - Mozilla Firefox
Nitos Experimental Por...
localhost:3000/reservation
NITOS
Home
News
Documentation
NITLab
About
Account
Home
Experimentation tools
Scheduler
Node Status
My Reservations
Select Date
ardadouk
Make reservations
2014 NITOS Experimental Portal by NITlab

Successful reservation!

Scheduler

Reservation table for the next 24 hours

Date	2014-11-11												2014-11-12													
	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30
node016																										
node002																										
node120								🔒	🔒	🔒																
node121																										
channel1																										
channel2																										

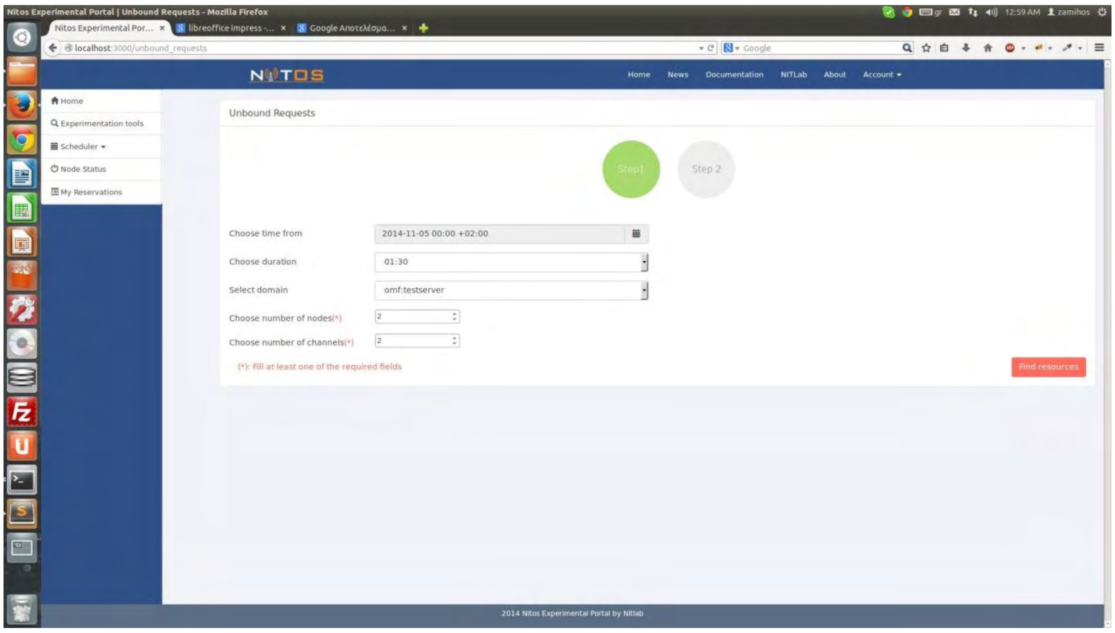
Timezone	+0200
🔒	Reserved Timeslots
✓	Checked

Figure 39: Reservation Tool – Successful Reservation

3.3.2.2 Quick Reservation Tool

The second tool we developed was Quick Reservation tool. This tool offers to the experimenters a different way for reserving resources. Unlike the Reservation tool where user has to discover the available resources, here, Quick Reservation tool does the job.

Once an experimenter enters the tool, he has to complete a form. He should give the number of nodes and/or the number of channels which he wants to reserve. If none of the nodes number or the channels number field will be completed, a relative message will be displayed informing the user that he has to complete at least one of the two fields. User also has the opportunity to define the start time for the reservation, the duration (until 4 hours per request) and the domain of the resource. If none of the above will be completed, tool will search from the next half hour and for a 1 hour period. After submitting the form, Quick Reservation tool will return some results. If there are available resources which match to the given criteria, a table with the resources will be displayed with an option for reserving the resources. If not, a message will inform user that there are not any resources that suits to the request.

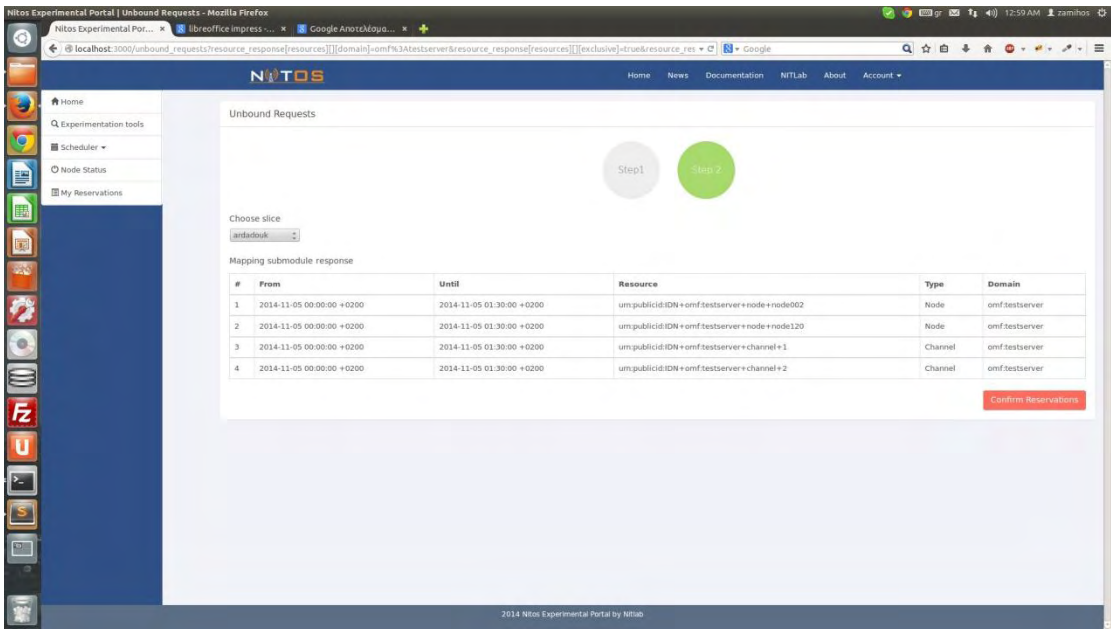


Unbound Requests

Choose time from	2014-11-05 00:00 +02:00
Choose duration	01:30
Select domain	omf:testserver
Choose number of nodes(*)	2
Choose number of channels(*)	2

(*): Fill at least one of the required fields

Figure 40: Quick Reservation – Filling the form



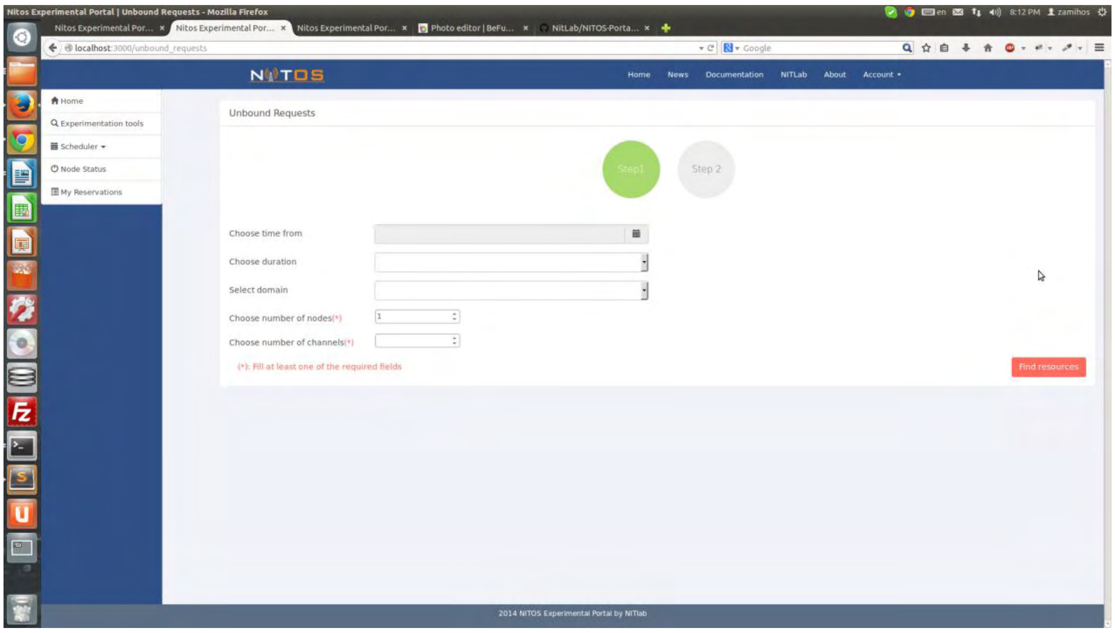
Unbound Requests

Choose slice: ardadouk

Mapping submodule response

#	From	Until	Resource	Type	Domain
1	2014-11-05 00:00:00 +0200	2014-11-05 01:30:00 +0200	urn:publicid:IDN+omf:testserver+node+node002	Node	omf:testserver
2	2014-11-05 00:00:00 +0200	2014-11-05 01:30:00 +0200	urn:publicid:IDN+omf:testserver+node+node120	Node	omf:testserver
3	2014-11-05 00:00:00 +0200	2014-11-05 01:30:00 +0200	urn:publicid:IDN+omf:testserver+channel+1	Channel	omf:testserver
4	2014-11-05 00:00:00 +0200	2014-11-05 01:30:00 +0200	urn:publicid:IDN+omf:testserver+channel+2	Channel	omf:testserver

Figure 41: Quick Reservation Tool– Returned results

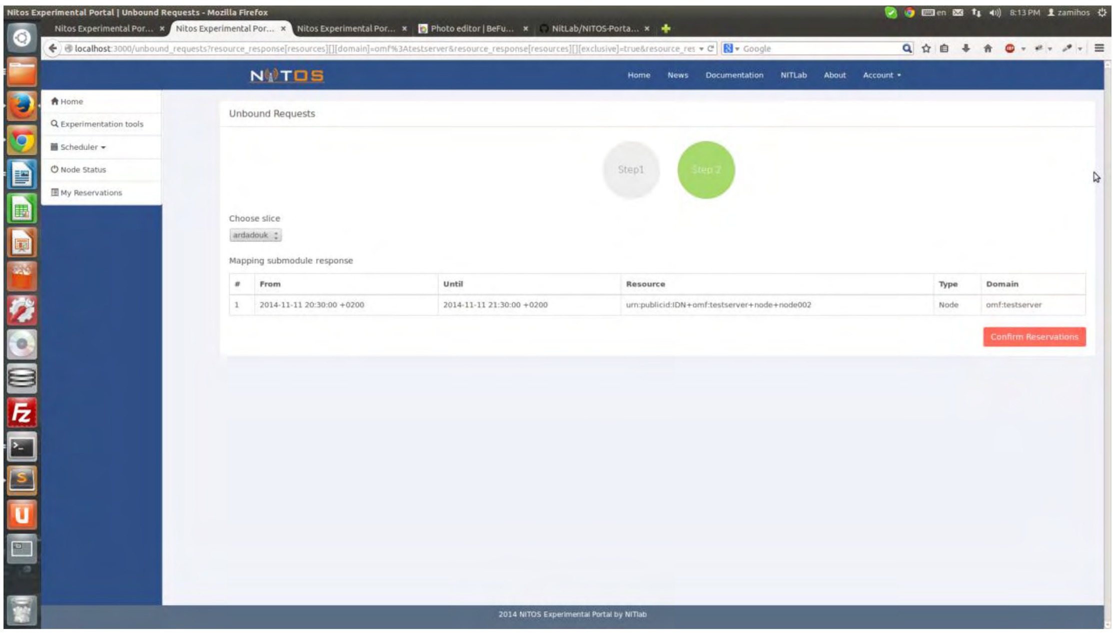

Nitos Experimental Portal | Unbound Requests - Mozilla Firefox
localhost:3000/unbound_requests
Home
News
Documentation
NITLab
About
Account
Home
Experimentation tools
Scheduler
Node Status
My Reservations
Step1
Step 2
Find resources
2014 NITOS Experimental Portal by NITlab

Unbound Requests

Choose time from	
Choose duration	
Select domain	
Choose number of nodes(*)	1
Choose number of channels(*)	

(*): Fill at least one of the required fields

Figure 42: Quick Reservation Tool - Select only number of resources



Unbound Requests

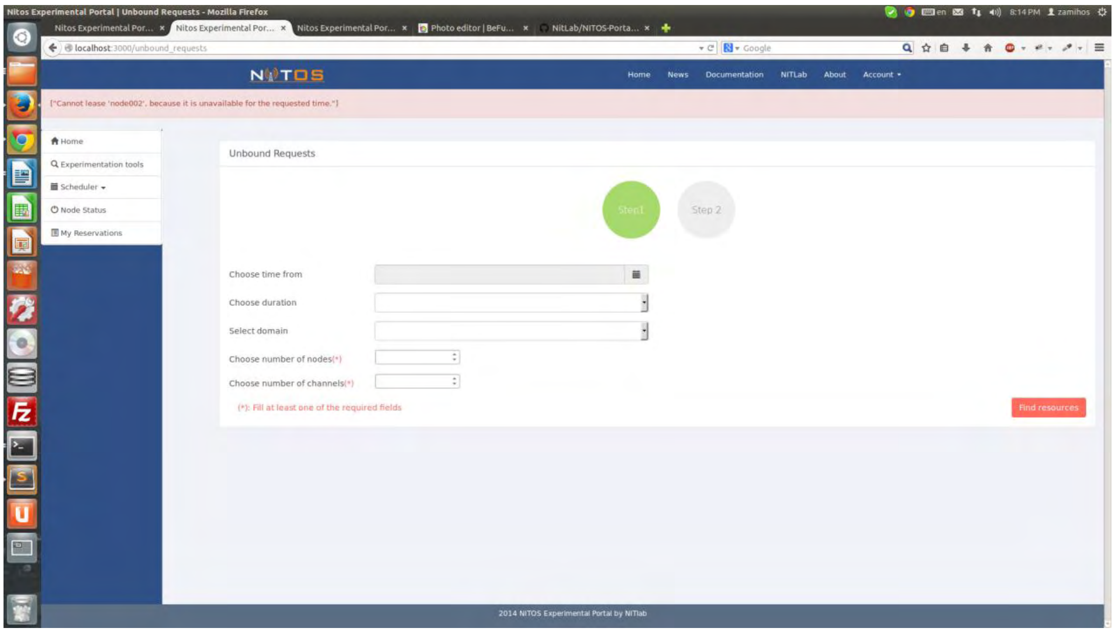
Choose slice: ardadouk

Mapping submodule response

#	From	Until	Resource	Type	Domain
1	2014-11-11 20:30:00 +0200	2014-11-11 21:30:00 +0200	urn:publicid:IDN+omf:testserver+node+node002	Node	omf:testserver

Figure 43: Quick Reservation Tool – Returned results for the next half hour and for 1 hour period

Like Reservation tool, a scenario of a failed reservation is when another user reserve the same resources just before us. Then a message will be displayed which informs us that the resources are already reserved and urging us trying again.


["Cannot lease 'node002', because it is unavailable for the requested time."]

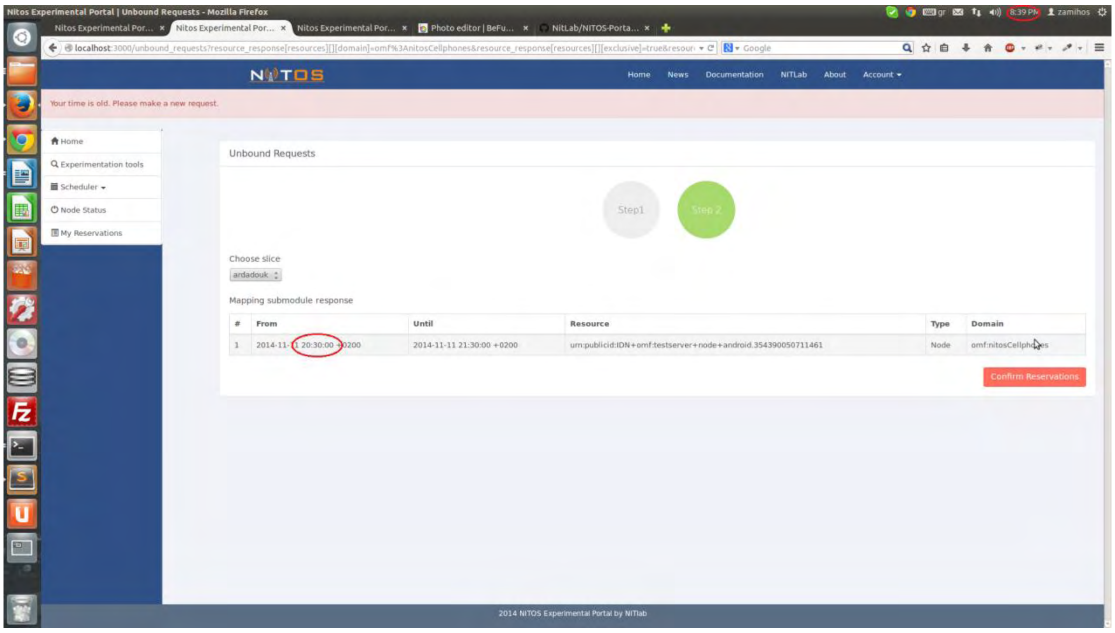
Unbound Requests

Choose time from	
Choose duration	
Select domain	
Choose number of nodes(*)	
Choose number of channels(*)	

(*): Fill at least one of the required fields

Figure 44: Quick Reservation Tool – Failure because of other user

Another scenario is with the forgotten time slots. If we try to make a reservation from a past time slot an error message will be displayed.


Your time is old. Please make a new request.

Unbound Requests

Choose slice: ardadouk

Mapping submodule response

#	From	Until	Resource	Type	Domain
1	2014-11-11 20:30:00 +0200	2014-11-11 21:30:00 +0200	urn:publicid:IDN+omf:testserver+node+android.354390050711461	Node	omf:nitosCellphones

Figure 45: Quick Reservation Tool – Forgotten time slots case

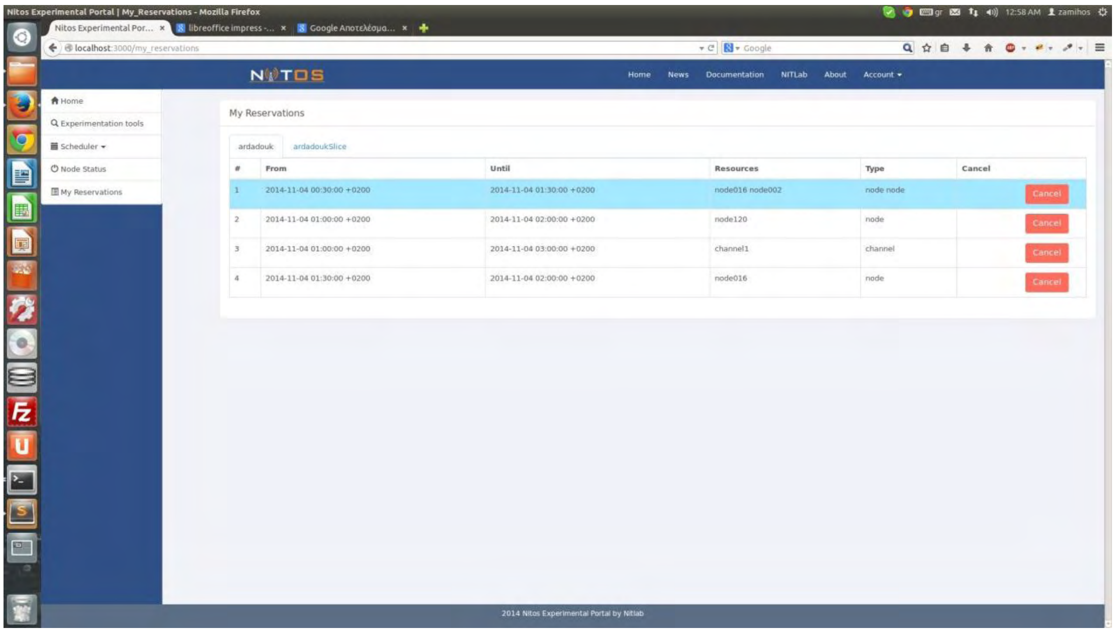
Similar as the Reservation tool, Quick Reservation displays all the information accordingly to the user's timezone. When a user makes a request, we make a GET request to the Broker. Broker searches for available resources that match the requests criteria and responds to the back-end. When a user makes a reservation, a POST request is sent to Broker for reserving the corresponding resources.

3.3.3 My Reservations Tool

As we saw in the previous chapters the user interface that we built allows users to reserve resources of NITOS. So we had to develop a tool for managing these reservations. This motivated us to develop the My Reservations tool. My Reservation gives user the opportunity to observe his/her reservations. If user reserved some resources and the reservation is active or will be active in future, tool will display the reservation in a table. We use a tabbed menu for displaying the reservations of every user's slice. The table doesn't keeps info for reservations that have ended. We display the active reservations (the current time is among the period time of the reservation) with a different color, making easier for the experimenter to perceive which are actives.

My Reservation tool lets user, also, to cancel a reservation. User may cancel a reservation whether it is active or not, so the resources will be available for reserving by other users from the next half hour.

The communication schema is similar to the schemes of the previous tools. The back-end of Portal communicates with Broker and shows all the information according to the user's timezone. When the user presses the cancel button a delete request is sent to the Broker. The figures below shows the My Reservation tool.

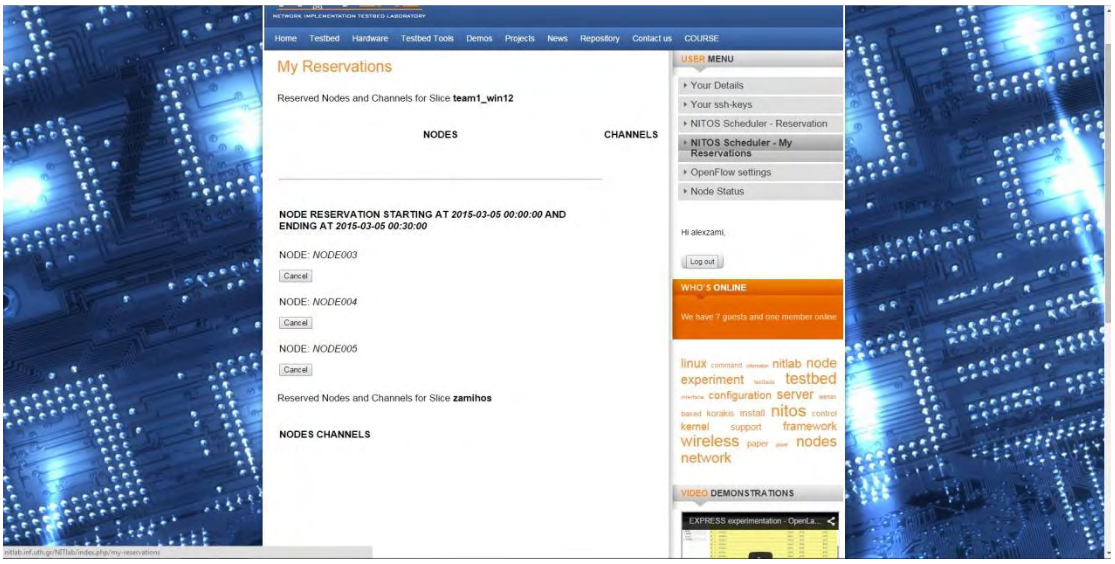

Nitos Experimental Portal | My_Reservations - Mozilla Firefox
localhost:3000/my_reservations
Home
News
Documentation
NITLab
About
Account
Home
Experimentation tools
Scheduler
Node Status
My Reservations
2014 Nitos Experimental Portal by Nitlab

My Reservations

ardadouk | ardadoukSlice

#	From	Until	Resources	Type	Cancel
1	2014-11-04 00:30:00 +0200	2014-11-04 01:30:00 +0200	node016 node002	node node	Cancel
2	2014-11-04 01:00:00 +0200	2014-11-04 02:00:00 +0200	node120	node	Cancel
3	2014-11-04 01:00:00 +0200	2014-11-04 03:00:00 +0200	channel1	channel	Cancel
4	2014-11-04 01:30:00 +0200	2014-11-04 02:00:00 +0200	node016	node	Cancel

Figure 46: My Reservation Tool Interface


NETWORK IMPLEMENTATION TESTBED LABORATORY
Home
Testbed
Hardware
Testbed Tools
Demos
Projects
News
Repository
Contact us
COURSE
USER MENU
Your Details
Your ssh-keys
NITOS Scheduler - Reservation
NITOS Scheduler - My Reservations
OpenFlow settings
Node Status
Hi alexzami,
Log out
WHO'S ONLINE
We have 7 guests and one member online
linux command nitlab node experiment testbed configuration server based korakis install nitos kernel support framework control wireless paper nodes network
VIDEO DEMONSTRATIONS
EXPRESS experimentation - OpenLa...

My Reservations

Reserved Nodes and Channels for Slice **team1_win12**

NODES CHANNELS

NODE RESERVATION STARTING AT *2015-03-05 00:00:00* AND ENDING AT *2015-03-05 00:30:00*

NODE: *NODE003*
Cancel

NODE: *NODE004*
Cancel

NODE: *NODE005*
Cancel

Reserved Nodes and Channels for Slice **zamihos**

NODES CHANNELS

nitlab.inf.uth.gr/NITlab/index.php/my-reservations

Figure 47: Previous Version of My Reservations Tool

3.3.4 Node Status tool

The last tool that we dealt with was Node Status tool, which is an integrated web-based tool for controlling and monitoring the status of the nodes according to a user's reservation. Figure illustrates the developed application where the available nodes are shown to the experimenter along with the appropriate keys to turn on/off or reset the reserved nodes. The non-reserved nodes cannot be turned on/off or reset, since this would lead to inappropriate use of resources.

As in My Reservation tool, we use a tabbed menu which categorizes all the information in the appropriate slice. We color the non-reserved and the non-active nodes with gray. A reserved-active node may be colored with red or green. If a node is closed we use red color, while for an opened we use green.

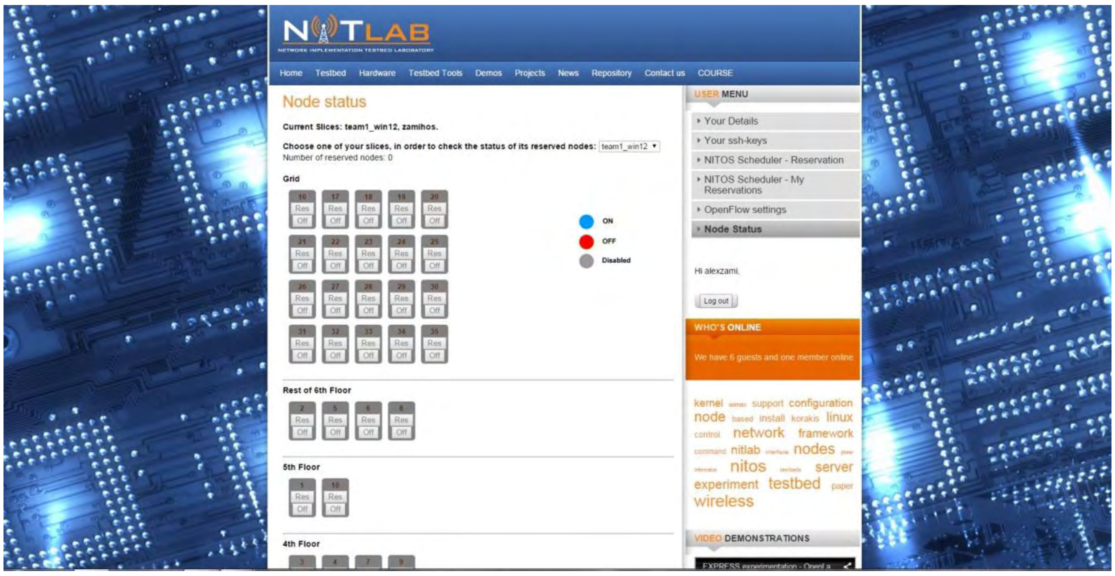


Figure 48: Previous Version of My Node Status Tool

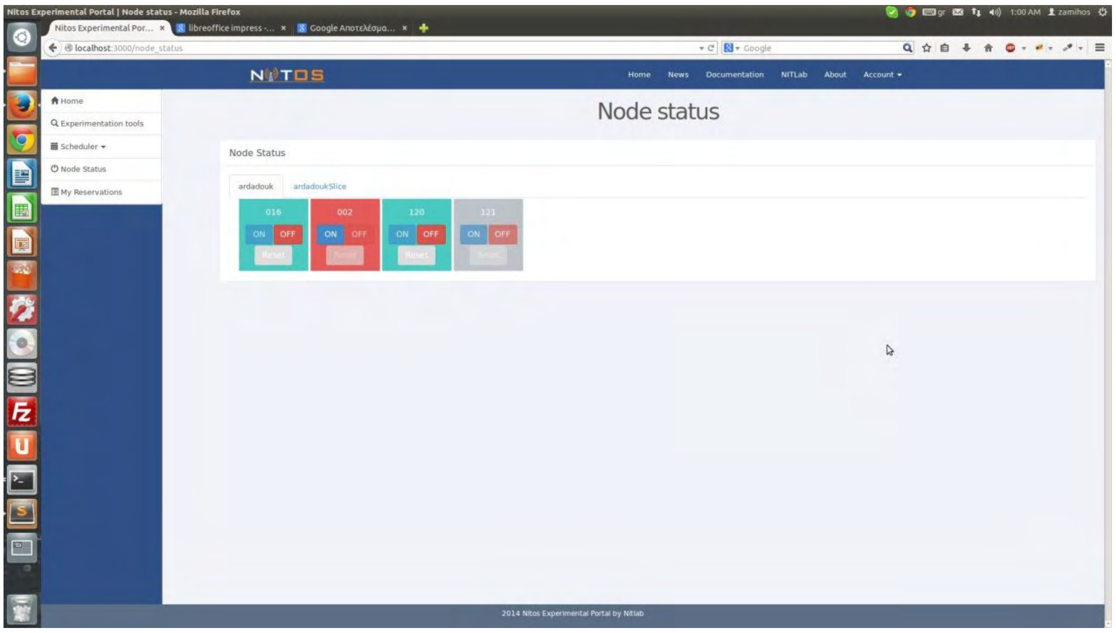


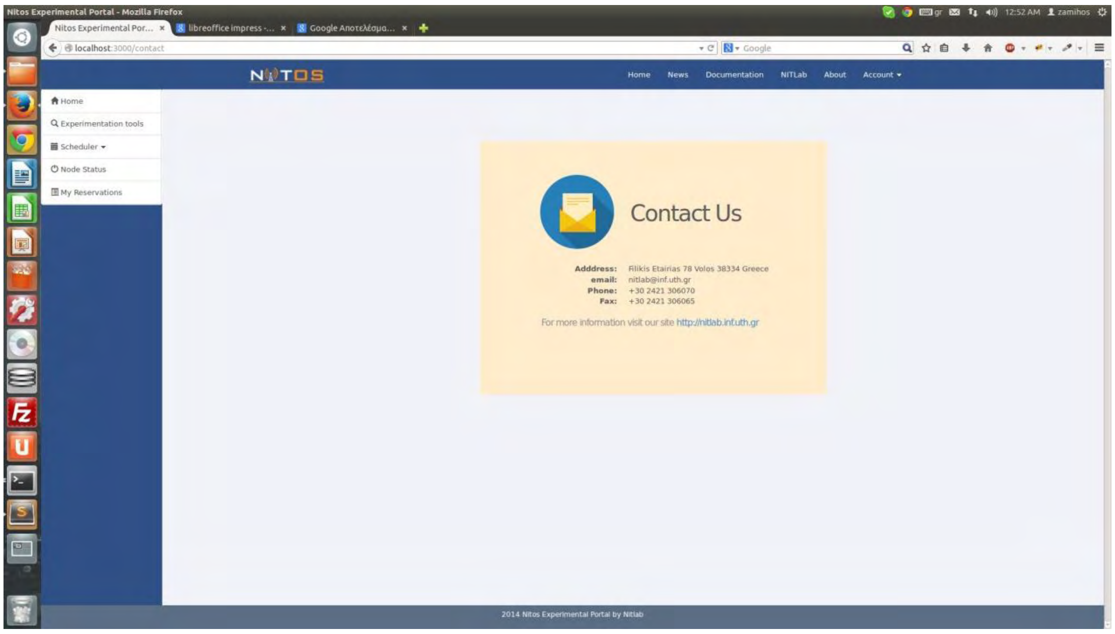
Figure 49: Node Status Tool Interface

Users have the opportunity to turn on/off or reset the nodes, by clicking the corresponding buttons. These send an HTTP request to the CM card. The CM card's HTTP response is immediate. In the case of the 'off' command the response may last from a few milliseconds to 8 seconds, until the node is switched off. In order to avoid client timeout or the CM card to appear as non-responsive for that period of time, we chose to execute the on/off and reset commands asynchronously.

For informing the front-end, we built a mechaninm based on websockets, using websockets-rails gem. We run a separate thread in the back-end, which asks every 2 seconds the CM cards for the status of all nodes. Websockets are used for the communication between back-end and front-end. When the back-end receives a response from the CM cards, an event triggered to the front-end. Then, we updates accordingly the information monitoring to the users.

3.3.5 Other Features

In this section we display some basic information that NITOS Portal provides to the experimenters. Also, we demonstrate the responsive design, which allows the Portal to be used on small screens, tablets and cell phones.



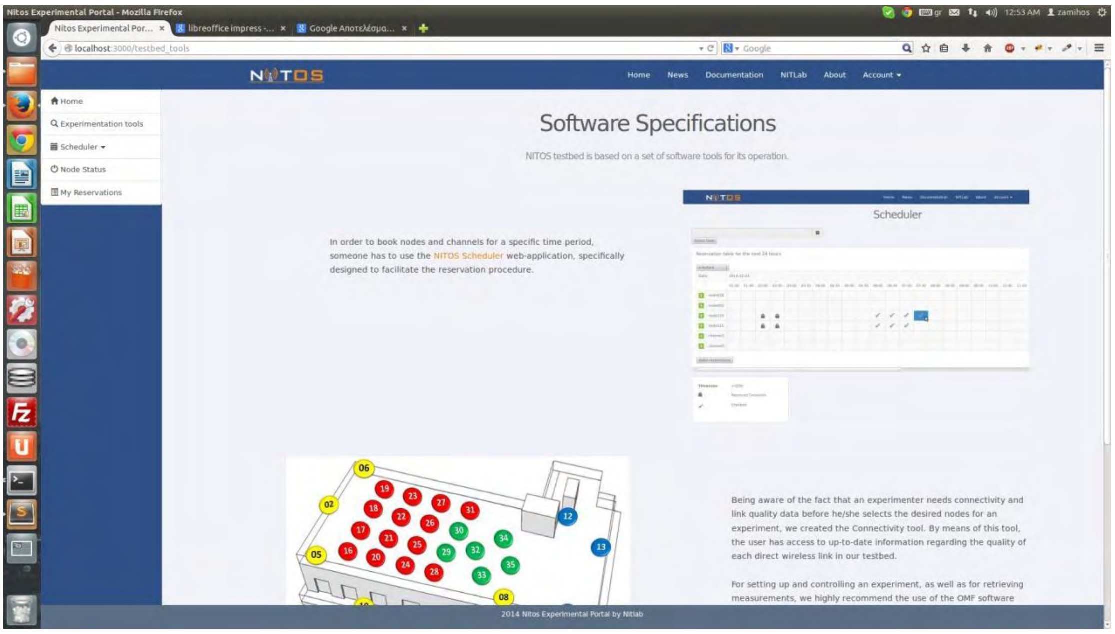
Contact Us

Adddress: Filikis Etairias 78 Volos 38334 Greece
email: nitlab@inf.uth.gr
Phone: +30 2421 306070
Fax: +30 2421 306065

For more information visit our site http://nitlab.inf.uth.gr

2014 Nitos Experimental Portal by Nitlab

Figure 50: Contact Page



Software Specifications

NITOS testbed is based on a set of software tools for its operation.

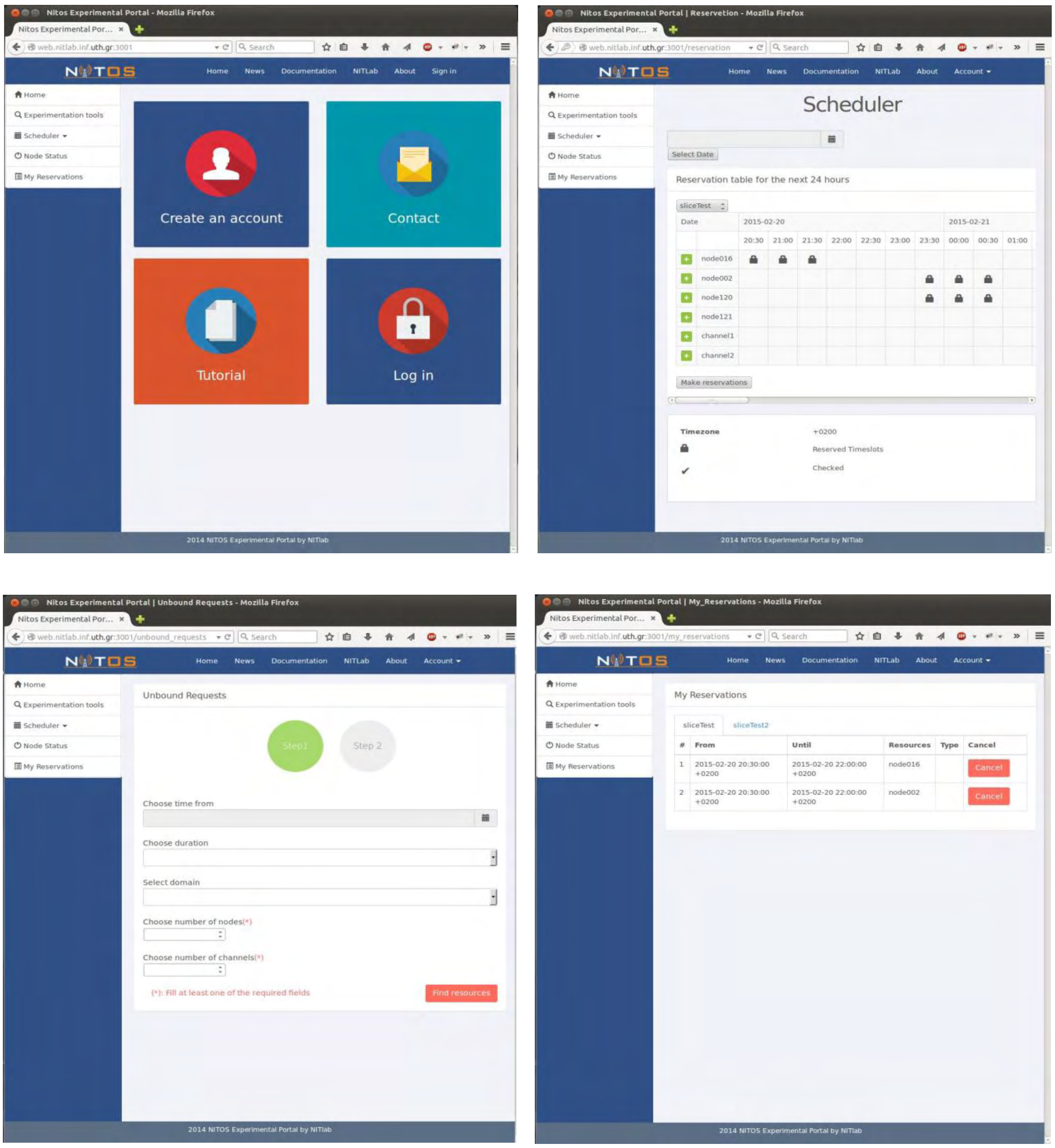
In order to book nodes and channels for a specific time period, someone has to use the NITOS Scheduler web-application, specifically designed to facilitate the reservation procedure.

Being aware of the fact that an experimenter needs connectivity and link quality data before he/she selects the desired nodes for an experiment, we created the Connectivity tool. By means of this tool, the user has access to up-to-date information regarding the quality of each direct wireless link in our testbed.

For setting up and controlling an experiment, as well as for retrieving measurements, we highly recommend the use of the OMF software

2014 Nitos Experimental Portal by Nitlab

Figure 51: Information Page


Nitos Experimental Portal - Mozilla Firefox
web.nitlab.inf.uth.gr:3001
NITOS
Home
News
Documentation
NITLab
About
Sign in
Experimentation tools
Scheduler
Node Status
My Reservations
Create an account
Contact
Tutorial
Log in
2014 NITOS Experimental Portal by NITlab
Nitos Experimental Portal | Reservation - Mozilla Firefox
web.nitlab.inf.uth.gr:3001/reservation
Account
Scheduler
Select Date
sliceTest
Make reservations
Nitos Experimental Portal | Unbound Requests - Mozilla Firefox
web.nitlab.inf.uth.gr:3001/unbound_requests
Step1
Step 2
Find resources
Nitos Experimental Portal | My_Reservations - Mozilla Firefox
web.nitlab.inf.uth.gr:3001/my_reservations
sliceTest
sliceTest2
Cancel

Reservation table for the next 24 hours

Date	2015-02-20							2015-02-21		
	20:30	21:00	21:30	22:00	22:30	23:00	23:30	00:00	00:30	01:00
node016	🔒	🔒	🔒							
node002							🔒	🔒	🔒	
node120							🔒	🔒	🔒	
node121										
channel1										
channel2										

Timezone	+0200
🔒	Reserved Timeslots
✓	Checked

Unbound Requests

Choose time from

Choose duration

Select domain

Choose number of nodes(*)

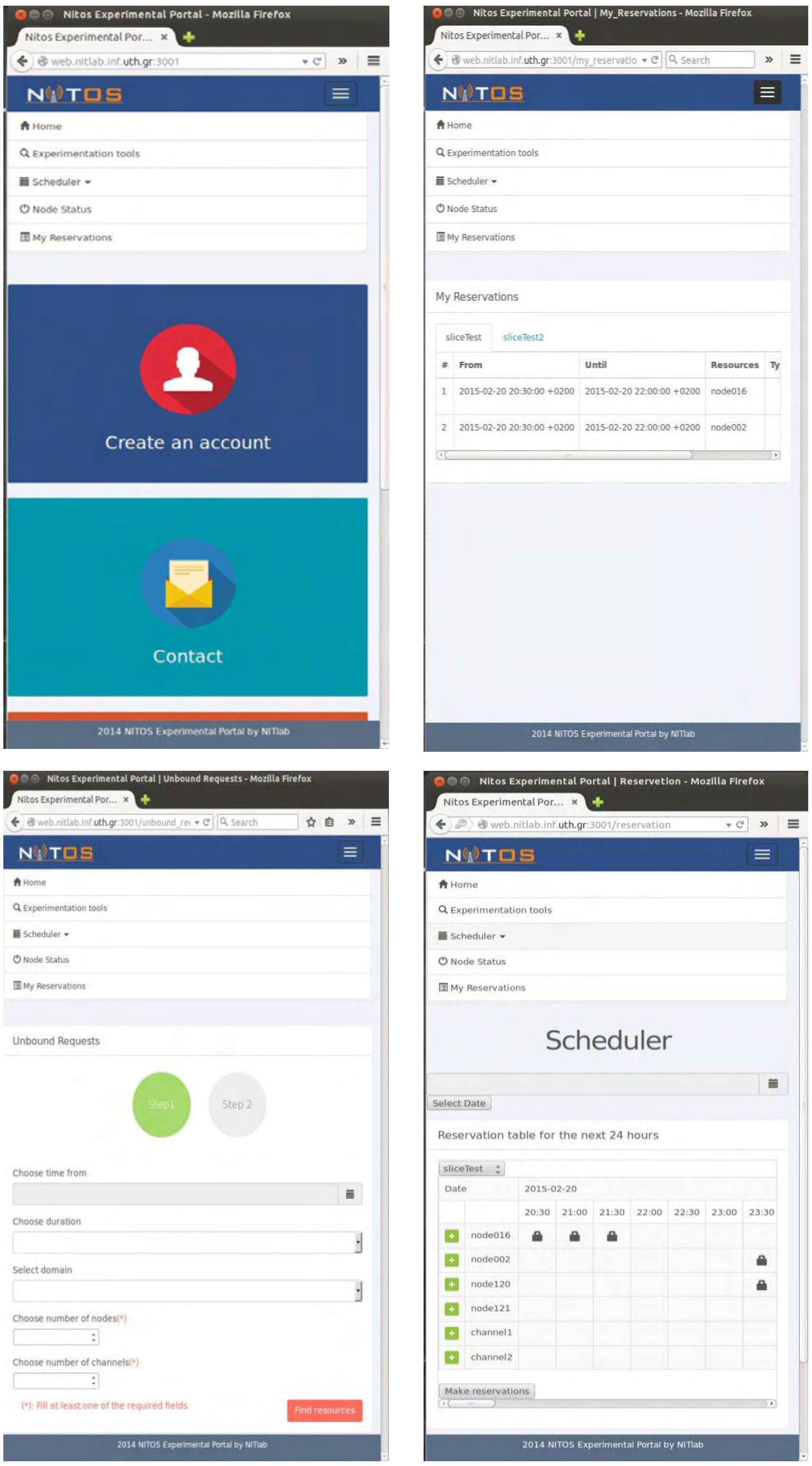
Choose number of channels(*)

(*): Fill at least one of the required fields

My Reservations

#	From	Until	Resources	Type	Cancel
1	2015-02-20 20:30:00 +0200	2015-02-20 22:00:00 +0200	node016		Cancel
2	2015-02-20 20:30:00 +0200	2015-02-20 22:00:00 +0200	node002		Cancel

Figure 52: PORTAL on Tablet's screen


Nitos Experimental Portal - Mozilla Firefox
web.nitlab.inf.uth.gr:3001
NITOS
Home
Experimentation tools
Scheduler
Node Status
My Reservations
Create an account
Contact
2014 NITOS Experimental Portal by NITlab
Nitos Experimental Portal | My_Reservations - Mozilla Firefox
web.nitlab.inf.uth.gr:3001/my_reservatio
My Reservations
sliceTest
sliceTest2
Nitos Experimental Portal | Unbound Requests - Mozilla Firefox
Unbound Requests
Step1
Step 2
Choose time from
Choose duration
Select domain
Choose number of nodes(*)
Choose number of channels(*)
(*): Fill at least one of the required fields
Find resources
Nitos Experimental Portal | Reservetion - Mozilla Firefox
web.nitlab.inf.uth.gr:3001/reservation
Scheduler
Select Date
Reservation table for the next 24 hours
sliceTest
Make reservations

#	From	Until	Resources	Ty
1	2015-02-20 20:30:00 +0200	2015-02-20 22:00:00 +0200	node016	
2	2015-02-20 20:30:00 +0200	2015-02-20 22:00:00 +0200	node002	

Date	2015-02-20						
	20:30	21:00	21:30	22:00	22:30	23:00	23:30
node016	🔒	🔒	🔒				
node002							🔒
node120							🔒
node121							
channel1							
channel2							

Figure 53: Portal on cell phone's screen

Conclusion

In conclusion, our aim was to create a simple, modern and easily expandable Portal which can be displayed on various screens (small screens, tablets, cell phones) for gathering the experimentation tools of NITOS. We managed to develop from scratch the already existed basic tools (Reservation, My Reservations and Node Status tools), making them to communicate with Broker and also added the new Quick Reservation tool. We also developed a Portal that can be easily expanded with new experimentation tools of NITOS.

References

1. NITOS Portal, https://github.com/NitLab/NITOS-Portal

2. OpenLab Deliverable D3.2: "Federation of wireless-specific tools and methodologies", http://www.ict-openlab.eu/fileadmin/documents/public_deliverables/OpenLab_Deliverable_D3_2.pdf

3. OpenLab Deliverable D3.3: "Provision of wireless testbeds", http://www.ict-openlab.eu/fileadmin/documents/public_deliverables/OpenLab_Deliverable_D3_3_updated.pdf

4. Michael Hartl, "The Ruby on Rails Tutorial", http://rails-4-0.railstutorial.org/book

5. Scott Nelson, Clientside Timezone Detection

6. Bootstrap, http://getbootstrap.com/

7. Site of NITlab, http://nitlab.inf.uth.gr/NITlab/

8. Vysakh Sreeenivasan, Websocket in Rails4