



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μοντέλα Περιγραφής και Διαχείρισης Πόρων σε Ασύρματες Πειραματικές Πλατφόρμες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΕΥΡΩΠΙΔΗ Χ. ΠΑΡΑΣΚΕΥΑ

Επιβλέπων: Συμεών Παπαβασιλείου
Αναπληρωτής Καθηγητής Ε.Μ.Π.

ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΙ ΒΕΛΤΙΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΔΙΚΤΥΩΝ
ΤΗΛΕΜΑΤΙΚΗΣ
(NETMODE)

Αθήνα, Ιούλιος 2011



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΙ ΒΕΛΤΙΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΔΙΚΤΥΩΝ
ΤΗΛΕΜΑΤΙΚΗΣ
(NETMODE)

Μοντέλα Περιγραφής και Διαχείρισης Πόρων σε Ασύρματες Πειραματικές Πλατφόρμες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
του
ΕΥΡΥΠΙΔΗ Χ. ΠΑΡΑΣΚΕΥΑ

Επιβλέπων: Συμεών Παπαβασιλείου
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 15^η Ιουλίου 2011.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Σ. Παπαβασιλείου
Αν.Καθηγητής Ε.Μ.Π.

.....
Β. Μάγκλαρης
Καθηγητής Ε.Μ.Π.

.....
Μ. Θεολόγου
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2011



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΙ ΒΕΛΤΙΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΔΙΚΤΥΩΝ
ΤΗΛΕΜΑΤΙΚΗΣ
(NETMODE)

(Υπογραφή)

.....
ΕΥΡΙΠΙΔΗΣ Χ. ΠΑΡΑΣΚΕΥΑΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © ΕΥΡΙΠΙΔΗΣ Χ. ΠΑΡΑΣΚΕΥΑΣ, 2011.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω όλους όσους συνέβαλαν με οποιονδήποτε τρόπο στην επιτυχή εκπόνηση αυτής της διπλωματικής εργασίας. Θα πρέπει να ευχαριστήσω θερμά τον καθηγητή κ. Συμεών Παπαβασιλείου για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο Διαχείρισης και Βέλτιστου Σχεδιασμού Δικτύων Τηλεματικής (NETMODE). Ήταν πάντα διαθέσιμος να μου προσφέρει τις γνώσεις και την εμπειρία του για την επιτυχή εκπόνηση διπλωματικής εργασίας και τις συμβουλές του για την διαδικασία των αιτήσεων για την συνέχιση των σπουδών μου στις Η.Π.Α.

Στη συνέχεια θα ήθελα να ευχαριστήσω ιδιαίτερα τους διδάκτορες κ. Χρύσα Παπαγιάννη και κ. Γιώργο Ανδρουλιδάκη για την εξαιρετική συνεργασία που είχαμε. Ήταν πάντα διαθέσιμοι να ασχοληθούν με οποιαδήποτε απορία είχα σχετικά με την διπλωματική εργασία. Οι επιστημονικές γνώσεις και οι πολύτιμες συμβουλές τους συνέβαλλαν τα μέγιστα στο να ολοκληρωθεί επιτυχώς η παρούσα διπλωματική εργασία.

Επιπλέον θα ήθελα να ευχαριστήσω τους καθηγητές της σχολής ΗΜΜΥ του Εθνικού Μετσόβιου Πολυτεχνείου που με καθοδήγησαν όλα αυτά τα χρόνια στο πολύ ενδιαφέρον αντικείμενο του ηλεκτρολόγου μηχανικού και μου πρόσφεραν χωρίς ενδοιασμό τις συστατικές τους επιστολές. Ιδιαίτερα θα ήθελα να ευχαριστήσω τον καθηγητή κ. Παναγιώτη Κωττή που μου δίδαξε βασικότατα μαθήματα στον τομέα των τηλεπικοινωνιών και των ασύρματων συστημάτων καθώς και τον καθηγητή κ. Μιχαήλ Θεολόγου.

Ακόμα ευχαριστώ τους φίλους μου για τη στήριξη που μου παρείχαν όλα αυτά τα χρόνια και τη συμπαράσταση τους στις δύσκολες στιγμές. Τέλος θα ήθελα να αφιερώσω την διπλωματική μου εργασία στους γονείς μου, Χρήστο και Ελένη, για την καθοδήγηση, τη συμπαράσταση και την εμπιστοσύνη που μου πρόσφεραν καθόλη τη διάρκεια των σπουδών μου.

Περίληψη

Η ανάπτυξη και χρήση πειραματικών πλατφορμών διευκολύνει την ανάπτυξη νέων αρχιτεκτονικών και πρωτοκόλλων για το Διαδίκτυο του μέλλοντος, αφού παρέχουν τους απαιτούμενους πόρους για διεξαγωγή πειραμάτων υπο πραγματικές συνθήκες. Για τη διαχείριση μιας πειραματικής πλατφόρμας απαιτείται ένας κοινός τρόπος περιγραφής των πόρων, ο οποίος θα είναι ανεξάρτητος της εκάστοτε υλοποίησης τόσο σε υλικό όσο και σε λογισμικό.

Στη διπλωματική εργασία δημιουργήσαμε μια επέκταση στη δομή δεδομένων Resource Specification (version 2), η οποία αποτελεί ένα μηχανισμό δημοσίευσης διαθέσιμων πόρων, περιγραφής αιτήσεων χρηστών για συγκεκριμένους πόρους και περιγραφή των πόρων οι οποίοι έχουν δεσμευθεί και χρησιμοποιούνται από τους ερευνητές, και ο οποίος αναπτύχθηκε από το ProtoGENI (www.protogeni.net). Η επέκταση χρησιμοποιήθηκε για την κατάλληλη περιγραφή των ασύρματων πόρων της πειραματικής πλατφόρμας του εργαστηρίου Διαχείρισης και Βέλτιστου Σχεδιασμού Δικτύων Τηλεματικής.

Στο δεύτερο μέρος της διπλωματικής εργασίας σχεδιάσαμε ένα σύστημα διαχείρισης για την πειραματική πλατφόρμα ασύρματων πόρων του εργαστηρίου Διαχείρισης και Βέλτιστου Σχεδιασμού Δικτύων Τηλεματικής, το οποίο κάνει χρήση της εκτεταμένης δομής RSpec v.2. Συγκεκριμένα το σύστημα αυτό επιτρέπει στους ερευνητές να δημιουργήσουν, να δεσμεύσουν και να διαχειριστούν τις φέτες (slices) πόρων τους. Με την έννοια φέτα πόρων (slice) εννοούμε ένα υποσύνολο των πόρων της εργαστηριακής πλατφόρμας τους οποίους δεσμεύει ο ερευνητής για τα πειράματα του. Η λειτουργία του συστήματος βασίζεται στο σχεδιασμό κατάλληλων υπηρεσιών διαδικτύου που περιέχουν τις απαιτούμενες λειτουργίες για δημιουργία, δέσμευση και διαχείριση των φετών. Παράλληλα αναπτύξαμε ένα χρονοπρογραμματιστή δέσμευσης πόρων ώστε οι ερευνητές να μπορούν να δεσμεύσουν μόνο τους διαθέσιμους πόρους σε κάποιο συγκεκριμένο χρονικό διάστημα. Τέλος σχεδιάσαμε ένα γραφικό περιβάλλον χρήστη για να μπορούν οι διάφοροι ερευνητές να χρησιμοποιήσουν τις παραπάνω υπηρεσίες διαδικτύου και να διαχειριστούν τα slices που έχουν δημιουργήσει.

Λέξεις Κλειδιά

Φέτα Πόρων, Διαμοιραζόμενοι Πόροι, RSpec, Υπηρεσίες Διαδικτύου, Πειραματική Πλατφόρμα, Χρονοπρογραμματιστής

Abstract

The development and wide use of experimental testbeds facilitates the development of new innovating architectures and protocols for the Internet of the future, as well as providing the necessary resources to conduct experiments under real conditions. Management of the testbed's resources requires a common way to describe these resources, which will be independent of each hardware and software implementation.

In this diploma thesis we created an extension to the Resource Specification (version 2) mechanism for advertising, requesting and describing the resources used by experimenters developed by ProtoGENI (www.protogeni.net). The extension was made to describe properly wireless resources at the testbed of Network Management and Optimal Design Laboratory.

In the second part of this diploma thesis we designed a system for managing wireless resources at the testbed of Network Management and Optimal Design Laboratory, based on this extended RSpec v.2 format. Specifically, the proposed system enables researchers to create, reserve and manage their slices. A slice is a subset of the testbed's resources, which a researcher reserves in order to conduct his experiment. The system is based on the design of appropriate web services that contain the operations required for the creation, reservation and management of slices. In parallel we developed a time-based slice scheduler in order for researchers to be able to reserve only available resources at the requested timeframe. Finally we designed a Graphical User Interface to allow different researchers to use these Web Services and manage their slices.

Key Words

Slice, Shared Resources, RSpec, Web Services, Testbed, Scheduler

Πίνακας Περιεχομένων

1.Εισαγωγή	11
1.1 Πρόλογος	11
1.2 Αντικείμενο της διπλωματικής	11
1.3 Περιγραφή της πειραματικής πλατφόρμας του εργαστηρίου διαχείρισης και βέλτιστης σχεδίασης δικτύων Ε.Μ.Π.	12
1.4 Αρχιτεκτονική με τη χρήση φέτας πόρων (slice)	13
1.5 Σχετική Εργασία και Έρευνα	14
2.Βασικά Πρωτόκολλα και Τεχνολογίες	19
2.1 Η γλώσσα XML	19
2.1.1 Περιγραφή της γλώσσας XML	19
2.1.2 Πλεονεκτήματα της XML	20
2.2 XML Σχήμα (xsd)	21
2.3 Πεδίο Ονομάτων (namespace)	22
2.4 Relax NG	22
2.5 Namespace-based Validation Dispatching Language (NVDL)	23
2.6 Ενιαίο Όνομα Πόρου	24
2.7 Web Services	26
2.7.1 Περιγραφή των υπηρεσιών διαδικτύου (Web services)	26
2.7.2 Γιατί να χρησιμοποιήσουμε Web Services	27
2.7.3 Βασικά πρωτόκολλα για υπηρεσίες διαδικτύου (Web Services)	27
2.7.3.1 SOAP (Simple Object Access Protocol) – Πρωτόκολλο Μεταφοράς	27
2.7.3.2 WSDL (Web Service Description Language)	29
2.7.4 Περιγραφή WSO2 WSF extension for php web services	34
3.Μοντέλα Περιγραφής Διαμοιραζόμενων Πόρων	37
3.1 Resource Specification (RSpec)	37
3.1.1 Περιγραφή του RSpec	37
3.1.2 Πλεονεκτήματα του RSpec	38
3.1.3 Βασικό σχήμα του RSpec v.2 (Core RSpec) και κατηγορίες RSpec v.2	39
3.1.3.1 RSpec v.2 Request	39
3.1.3.2 RSpec v.2 Advertisement	41
3.1.3.3 RSpec v.2 Manifest	43
3.2 Επέκταση RSpec για περιγραφή ασύρματων κόμβων	45
3.2.1 Χαρακτηριστικά ασύρματων κόμβων	46
3.2.2 RSpec σχήμα επέκτασης	49
3.3 RSpec έλεγχος εγκυρότητας	55

4.Αρχιτεκτονική Συστήματος	57
4.1 Περιγραφή της Αρχιτεκτονικής	57
4.2 Περιγραφή της Βάσης Δεδομένων	59
5.Περιγραφή Υπηρεσιών Ιστού	65
5.1 Βασικές Έννοιες και Παραδοχές	65
5.1.1 Καταστάσεις των <i>slices</i>	65
5.1.2 Τύποι δεδομένων	67
5.2 Υπηρεσίες Ιστού (<i>Web Services</i>)	69
5.2.1 <i>loginservice</i> -Ταυτοποίηση Χρηστών.....	69
5.2.2 <i>mksliceservice</i> -Δημιουργία/Τροποποίηση <i>slice</i>	69
5.2.3 <i>reservesliceservice</i> -Κράτηση ενός <i>slice</i> για μελλοντική χρήση.....	71
5.2.4 <i>deactivateservice</i> -Απενεργοποίηση ενός <i>activated slice</i>	72
5.2.5 <i>cancelreservationservice</i> -Ακύρωση της κράτησης ενός <i>slice</i>	72
5.2.6 <i>delsliceservice</i> -Διαγραφή ενός <i>slice</i>	73
5.2.7 <i>listslicesservice</i> - Λίστα των <i>slices</i> και των διαθέσιμων πόρων	73
6.Περιγραφή Γραφικού Περιβάλλοντος Χρήστη (Graphical User interface)	81
7.Ανασκόπηση-Μελλοντική Εργασία	103
7.1 Ανασκόπηση	103
7.2 Μελλοντική Εργασία.....	103
Παράρτημα Α: Βιβλιογραφία	105
Παράρτημα Β: Κώδικας	108

Κεφάλαιο 1

Εισαγωγή

1.1 Πρόλογος

Η προσπάθεια που γίνεται τα τελευταία χρόνια για ενοποίηση ετερογενών πειραματικών πλατφορμών, που περιέχουν τόσο ενσύρματους όσο και ασύρματους πόρους, έχει δημιουργήσει την ανάγκη για δημιουργία ενός ενιαίου σχήματος περιγραφής των πόρων μιας πειραματικής πλατφόρμας. Το γεγονός αυτό θα δώσει την δυνατότητα στον ερευνητή για χρήση πόρων απο πολλές ετερογενείς πειραματικές πλατφόρμες. Προς την κατεύθυνση αυτή έχουν αναπτυχθεί διάφορες δομές δεδομένων απο ερευνητικές ομάδες ώστε να συμπεριλαμβάνονται ορισμένα βασικά χαρακτηριστικά για την περιγραφή των πόρων. Τέτοια χαρακτηριστικά είναι το αν είναι ο πόρος κόμβος ή ζεύξη, ο τύπος των διεπαφών που διαθέτει, η τοποθεσία του, τι υπηρεσίες υποστηρίζει και άλλα σημαντικά χαρακτηριστικά που θα αναλύσουμε στην παρούσα διπλωματική.

Στόχος αυτών των δομών δεδομένων είναι η δυνατότητα περιγραφής πόρων τόσο ενσύρματων όσο και ασύρματων και η δημιουργία κατάλληλου RSpec (Resource Specification) βασισμένου σε αυτές. Το RSpec, όπως αναφέρεται στο [1], αποτελεί ένα XML κείμενο το οποίο περιέχει τα απαραίτητα χαρακτηριστικά για την περιγραφή των πόρων. Το RSpec αρχείο επειδή έχει XML δομή είναι εύκολα διαχειρίσιμο τόσο απο το χρήστη όσο και απο τη διαχειριστική οντότητα της πειραματικής πλατφόρμας. Με αυτό τον τρόπο μπορεί να επιτευχθεί επικοινωνία μεταξύ του χρήστη και του εξυπηρετητή σχετικά με το ποιοι πόροι είναι διαθέσιμοι, τι πόρους χρειάζεται ο χρήστης και τι πόρους έχει ήδη δεσμεύσει.

1.2 Αντικείμενο της διπλωματικής

Στην παρούσα διπλωματική ασχοληθήκαμε με δύο βασικά ζητήματα. Το πρώτο είναι η επέκταση μιας ήδη υπάρχουσας δομής δεδομένων RSpec, με σκοπό την κατάλληλη περιγραφή ασύρματων πόρων. Το δεύτερο ζητούμενο της διπλωματικής είναι ο σχεδιασμός και η υλοποίηση κατάλληλων υπηρεσιών ιστού με στόχο τη δημοσιοποίηση των διαθέσιμων πόρων της πειραματικής πλατφόρμας στον χρήστη μέσω γραφικού περιβάλλοντος, καθώς και η ανάπτυξη εργαλείου scheduling για τα πειράματα των διάφορων ερευνητών.

Στο πρώτο μέρος βασιστήκαμε σε μια ήδη υπάρχουσα δομή πληροφορίας RSpec της εργαστηριακής ομάδας του ProtoGENI [2], η οποία παρέχει τη δυνατότητα για δημιουργία επεκτάσεων για νέους πόρους, προκειμένου να δημιουργήσουμε την κατάλληλη επέκταση για την περιγραφή των ασύρματων πόρων της εργαστηριακής μας πλατφόρμας. Συγκεκριμένα καθορίσαμε τα απαραίτητα χαρακτηριστικά για την

λειτουργία των ασύρματων διεπαφών (interfaces) των ασύρματων κόμβων της εργαστηριακής πλατφόρμας.

Στο δεύτερο μέρος της διπλωματικής, βασιζόμενοι στη δομή δεδομένων RSpec που επεκτείναμε, σχεδιάσαμε ένα σύστημα βασιζόμενο σε κάποιες υπηρεσίες ιστού (web services) με στόχο την δημιουργία ενός γραφικού περιβάλλοντος με το οποίο ο χρήστης θα μπορεί να διαχειρίζεται τα πειράματα του. Η λειτουργία του συστήματος βασίζεται στην δημιουργία και διαχείριση slices από το χρήστη για τα πειράματα που θέλει να εκτελέσει. Το slice, όπως θα εξηγήσουμε αναλυτικότερα παρακάτω, είναι ένα σύνολο από πόρους που δεσμεύει ο ερευνητής προκειμένου να δοκιμάσει καινούργιες τεχνολογίες και να εκτελέσει πειράματα. Ο χρήστης μπορεί να δημιουργήσει ένα ή περισσότερα slices και να δεσμεύσει πόρους σε αυτά. Παράλληλα όμως η ανάγκη για αξιόπιστη δέσμευση των πόρων της πειραματικής πλατφόρμας επιβάλλει την δημιουργία ενός προγραμματιστή δέσμευσης πόρων (scheduler), παρόμοιο με αυτόν που παρουσιάζεται στο [3]. Έτσι ο ερευνητής θα έχει τη δυνατότητα να δεσμεύσει σε κάποιο slice του μόνο τους πόρους που είναι διαθέσιμοι και δεν χρησιμοποιούνται από κάποιο «ενεργό» slice εκείνο το χρονικό διάστημα που θέλει. Το γεγονός αυτό επιτρέπει σε πολλούς χρήστες να χρησιμοποιούν τους πόρους της εργαστηριακής πλατφόρμας ταυτόχρονα χωρίς επικάλυψη. Ο κάθε χρήστης θα έχει τα δικά του slices με τους δικούς του δεσμευμένους πόρους χωρίς να επηρεάζει τα πειράματα των άλλων ερευνητών. Αξίζει να επισημάνουμε σε αυτό το σημείο ότι οι πόροι που διαμοιράζονται χρονικά στα διάφορα slices των χρηστών είναι οι ασύρματοι κόμβοι του εργαστηρίου και τα κανάλια μετάδοσης (συχνότητα) που ορίζουμε στα interfaces των κόμβων αυτών.

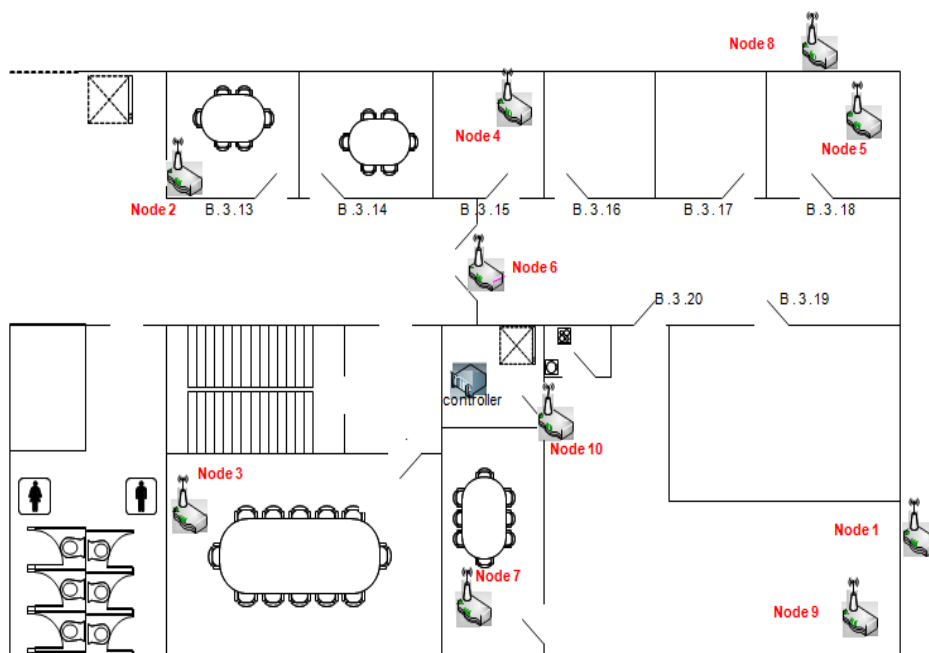
1.3 Περιγραφή της πειραματικής πλατφόρμας του εργαστηρίου διαχείρισης και βέλτιστης σχεδίασης δικτύων Ε.Μ.Π.

Τα τελευταία χρόνια έχουν αναπτυχθεί πολλές πειραματικές πλατφόρμες (testbeds) από ερευνητικές ομάδες ανά τον κόσμο, προκειμένου οι ερευνητές να χρησιμοποιούν τους διαθέσιμους πόρους τους για να πραγματοποιήσουν τα πειράματα τους, να δοκιμάσουν τις υπηρεσίες τους και να αναπτύξουν καινούργιες τεχνολογίες. Οι πειραματικές πλατφόρμες δύναται να αποτελούν μια μικρογραφία του πραγματικού Internet και προσφέρουν στους χρήστες μεγάλο πλήθος πόρων. Για αυτό το λόγο είναι πολύ σημαντικό εργαλείο για τους ερευνητές.

Στην παρούσα διπλωματική χρησιμοποιήσαμε την πειραματική πλατφόρμα του εργαστηρίου Διαχείρισης και Σχεδιασμού Δικτύων Τηλεματικής (NETMODE) για να δοκιμάσουμε τις υπηρεσίες και τα μοντέλα που αναπτύξαμε. Η πειραματική πλατφόρμα του εργαστηρίου αποτελείται από 10 ασύρματους κόμβους, οι οποίοι στην ουσία είναι pc engines τύπου alix2c2, τυχαία τοποθετημένοι στο χώρο του εργαστηρίου. Οι κόμβοι αυτοί διαχειρίζονται από το εργαλείο διαχείρισης OMF (cOntrol and Management Framework) [4]. Το OMF αναπτύχθηκε για τον έλεγχο και τη διαχείριση πειραματικών πλατφορμών και μάλιστα μπορεί να χρησιμοποιηθεί για

Κεφάλαιο 1:Εισαγωγή

διαχείριση σε testbeds, τα οποία περιλαμβάνουν τόσο ενσύρματους όσο και ασύρματους πόρους. Το μηχάνημα στο οποίο έχει εγκατασταθεί το εργαλείο διαχείρισης των ασύρματων πόρων OMF και το οποίο διαχειρίζεται τους 10 ασύρματους κόμβους του testbed είναι ένας Linux εξυπηρετητής (server) με DNS όνομα sorch.netmode.ntua.gr. Μια σχηματική αναπαράσταση της πειραματικής πλατφόρμας του εργαστηρίου φαίνεται στην παρακάτω εικόνα:



Σχήμα 1.1: Κάτοψη της πειραματικής πλατφόρμας του εργαστηρίου

Τα pc engines, που αποτελούν τους κόμβους του testbed, διαθέτουν επεξεργαστή 500 MHz AMD Geode LX800, μνήμη RAM 256 MB DDR DRAM και δύο ασύρματες διεπαφές για την επικοινωνία με τους υπόλοιπους ασύρματους κόμβους. Επιπλέον σε αυτούς τους κόμβους η διαχειριστική οντότητα φορτώνει ένα κοινό disk_image, το οποίο καθορίζει για παράδειγμα το λειτουργικό σύστημα που θα τρέξει στον συγκεκριμένο κόμβο.

Ο ερευνητής χρησιμοποιώντας τους πόρους της πειραματικής πλατφόρμας μπορεί να εκτελέσει τα πειράματα του υπό πραγματικές συνθήκες και να δοκιμάσει καινούργιες τεχνολογίες, πρωτόκολλα και υπηρεσίες.

1.4 Αρχιτεκτονική με τη χρήση φέτας πόρων (slice)

Τα τελευταία χρόνια οι περισσότερες ερευνητικές ομάδες προκειμένου να διεξάγουν πειράματα για νέες τεχνολογίες και νέα πρωτόκολλα για το Internet χρησιμοποιούν την λογική της δημιουργίας φετών πόρων από τους ερευνητές. Η ιδέα αυτή προήλθε από το πανεπιστήμιο του Princeton σε μια προσπάθεια αξιοποίησης των πόρων της πειραματικής πλατφόρμας Planetlab [5], που περιλαμβάνει περίπου 1000 κόμβους διασκορπισμένους σε διάφορες περιοχές του κόσμου.

Κεφάλαιο 1:Εισαγωγή

Με την έννοια φέτα πόρων (slice) εννοούμε ένα υποσύνολο των διαμοιραζόμενων πόρων της πειραματικής πλατφόρμας που δεσμεύει ο χρήστης για να δοκιμάσει τις υπηρεσίες και τα πρωτόκολλά του (όπως αναφέρεται στο [6]). Τέτοιοι πόροι είναι για παράδειγμα υπολογιστικοί κόμβοι, ζεύξεις, διεπαφές και κανάλια μετάδοσης (channels). Οι πόροι αυτοί μπορεί να ανήκουν σε μία ή περισσότερες πειραματικές πλατφόρμες εφ'όσον έχουν ενοποιηθεί μεταξύ τους (federated network testbeds) και έχουν ένα κοινό τρόπο περιγραφής των πόρων τους. Στην παρούσα διπλωματική οι πόροι που εκχωρούνται στο slice είναι ασύρματοι κόμβοι, στους οποίους κάνουμε αποκλειστική χρήση. Αποκλειστική χρήση γίνεται και στα κανάλια μετάδοσης των ασύρματων διεπαφών των κόμβων που εμπεριέχονται στο slice. Εξίσου σημαντικό είναι το γεγονός ότι σε ένα slice του ερευνητή μπορούν να εκχωρηθούν πόροι, οι οποίοι είναι διαθέσιμοι στο χρονικό διάστημα που επιθυμεί να το χρησιμοποιήσει. Αυτό συνιστά στην πιο αξιόπιστη χρήση των πόρων της πειραματικής πλατφόρμας, αφού ο ερευνητής κάθε φορά μπορεί να εκχωρήσει στο slice του μόνο τους διαθέσιμους πόρους και έτσι δεν επηρεάζει τα πειράματα που εκτελούνται από κάποιο άλλο «ενεργό» slice εκείνο το χρονικό διάστημα.

Επιπλέον η ιδέα της δημιουργίας των slices περιλαμβάνει και τον έλεγχο τους από μια ομάδα χρηστών. Ένα συγκεκριμένο slice θα μπορεί να διαχειρίζεται και να επεξεργάζεται από έναν ή περισσότερους χρήστες. Στην παρούσα διπλωματική όμως κάναμε τη σύμβαση ότι το slice θα χρησιμοποιείται μόνο από εκείνον τον χρήστη που το δημιούργησε και δεν θα δίνεται η δυνατότητα από τη διαχειριστική οντότητα για επεξεργασία από κάποιον άλλο χρήστη. Επομένως με την χρησιμοποίηση της slice-based αρχιτεκτονικής ο κάθε χρήστης δημιουργεί τα δικά του slices για τα πειράματα του εκχωρώντας σε αυτά διαθέσιμους πόρους, μπορεί να τα επεξεργαστεί όπως αυτός επιθυμεί και να τα δεσμεύσει πολλές φορές για μελλοντική χρήση εφόσον δεν υπάρχει χρονική επικάλυψη κάποιου κόμβου του slice ή κάποιου καναλιού που χρησιμοποιείται από κάποιο άλλο «ενεργό» slice.

1.5 Σχετική Εργασία και Έρευνα

Στο υποκεφάλαιο αυτό γίνεται ανασκόπηση των σχετικών εργασιών που έχουν γίνει σχετικά με το αντικείμενο της παρούσας διπλωματικής. Συγκεκριμένα θα εξετάσουμε τόσο τη σχετική έρευνα σχετικά με την περιγραφή των πόρων των πειραματικών πλατφορμών όσο και με την ανάπτυξη συστήματος διαχείρισης των ασύρματων πόρων της πειραματικής πλατφόρμας.

Περιγραφή πόρων των πειραματικών πλατφορμών

Στο κομμάτι της περιγραφής των πόρων έχουν αναπτυχθεί, όπως περιγράφεται στο [8], αρκετές γλώσσες περιγραφής για τους πόρους μίας πειραματικής πλατφόρμας. Αυτές είναι η Language for Embedded Networked Sensing (LENS) [8], η Network Description Language (NDL) [10], η επέκταση της NDL-OWL [11] και η Network Markup Language (NML) [12].

Κεφάλαιο 1:Εισαγωγή

Η LENS [8] είναι μια γλώσσα περιγραφής πόρων που καθορίζει μια σημασιολογική οντολογία για την περιγραφή των δικτύων ασύρματων αισθητήρων και αναπτύχθηκε από την ερευνητική ομάδα KanseiGenie [7]. Η οντολογία της LENS περιλαμβάνει όλες τις απαραίτητες έννοιες για την πλήρη περιγραφή των χαρακτηριστικών που καθορίζουν τη λειτουργία ενός ασύρματου δικτύου αισθητήρων (Wireless Sensor Network (WSN)).

Οι υπόλοιπες τρεις γλώσσες έχουν σχεδιαστεί κυρίως για την περιγραφή ενσύρματων δικτύων (wired networks) και βασίζονται σε οντολογίες. Η NDL [10] βασίζεται σε μια οντολογία για την περιγραφή ενσύρματων δικτύων. Η NDL-OWL [11] αποτελεί επέκταση της NDL γλώσσας που περιλαμβάνει νέες έννοιες για τον καθορισμό των δυνατοτήτων που προσφέρουν οι πειραματικές πλατφόρμες καθώς και το διαμοιρασμό των πόρων. Η NML [12] είναι μια γλώσσα που στοχεύει στην ανάπτυξη κατάλληλου επεκτάσιμου σχήματος για την περιγραφή υβριδικών δικτύων.

Επιπλέον για την περιγραφή των πόρων έχει αναπτυχθεί από το PlanetLab [5] και το ProtoGENI [2] η δομή δεδομένων RSpec (Resource Specification). Η δομή δεδομένων RSpec αποτελεί ένα XML κείμενο που περιλαμβάνει όλα τα απαραίτητα χαρακτηριστικά για την περιγραφή των πόρων των πειραματικών πλατφορμών.

Συστήματα διαχείρισης ασύρματων πόρων

Στο κομμάτι της διαχείρισης των ασύρματων πόρων της πειραματικής πλατφόρμας έχουν γίνει σχετικές εργασίες και έρευνα από διάφορες ερευνητικές ομάδες. Ενδεικτικά αναφέρουμε το εργαστήριο WinLAB [9], NITLab [13] (Network Implementation Testbed Laboratory) και την ερευνητική ομάδα Kansei Genie [7].

WINLAB for the ORBIT wireless network testbed

Το εργαστήριο WINLAB [9] ανέπτυξε ένα εργαλείο για διαχείριση πόρων, έλεγχο των πειραμάτων και των πειραματικών μετρήσεων, το OMF [4] (cOntrol and Management Framework). Το εργαλείο αυτό σχεδιάστηκε για να μπορεί να υποστηρίξει διαχείριση πόρων τόσο σε πειραματικές πλατφόρμες που περιέχουν ενσύρματους πόρους όσο και σε πειραματικές πλατφόρμες που αποτελούνται από ασύρματους πόρους. Επιπλέον μπορεί να χρησιμοποιηθεί σε πειραματικές πλατφόρμες που αποτελούν συνένωση ενσύρματων και ασύρματων πόρων.

Η αρχιτεκτονική του OMF αποτελείται από τρία πεδία, το πεδίο του ελέγχου (Control Plane), το πεδίο της διαχείρισης (Measurement Plane) και το πεδίο των μετρήσεων (Measurement Plane), όπως περιγράφεται στο [4]. Το πεδίο του ελέγχου περιλαμβάνει τα αντίστοιχα εργαλεία του OMF που χρησιμοποιούν οι ερευνητές για να περιγράψουν τα πειράματα τους καθώς και τις οντότητες του OMF για την πραγματοποίηση των πειραμάτων. Το πεδίο της διαχείρισης περιλαμβάνει τις συναρτήσεις του OMF, οι οποίες χρησιμοποιούνται για την δέσμευση των πόρων στους διάφορους ερευνητές αλλά και για καθορισμό των απαραίτητων χαρακτηριστικών των πόρων για την εκτέλεση των πειραμάτων. Τέλος το πεδίο των

Κεφάλαιο 1:Εισαγωγή

μετρήσεων περιλαμβάνει τα διάφορα εργαλεία του OMF για τη συλλογή και αποθήκευση των δεδομένων που προκύπτουν από τα πειράματα.

NITLab

Το εργαστήριο NITLab έχει αναπτύξει μια πειραματική πλατφόρμα αποτελούμενη από ασύρματους κόμβους (NITOS) με στόχο την διεξαγωγή πειραμάτων από τους ερευνητές σε ασύρματο περιβάλλον. Η λειτουργία της πειραματικής πλατφόρμας βασίζεται στην λειτουργία ενός κατάλληλου χρονικού προγραμματιστή πόρων (scheduler) για την καλύτερη αξιοποίηση της πειραματικής πλατφόρμας από τους διάφορους ερευνητές που τη χρησιμοποιούν. Με βάση το εργαλείο scheduling οι ασύρματοι πόροι διαμοιράζονται χρονικά στα διάφορα slices των ερευνητών και δεν υπάρχει παρεμβολή μεταξύ των διαφόρων πειραμάτων. Οι πόροι που διαμοιράζονται χρονικά μεταξύ των χρηστών είναι οι ασύρματοι κόμβοι και τα κανάλια μετάδοσης. Οι διάφοροι χρήστες δίνουν το χρονικό διάστημα που θέλουν να χρησιμοποιήσουν τους ασύρματους πόρους της εργαστηριακής πλατφόρμας και τους εμφανίζεται σε ένα γραφικό περιβάλλον ποιοί κόμβοι και ποία κανάλια είναι διαθέσιμα εκείνο το χρονικό διάστημα για να χρησιμοποιήσουν στα πειράματα τους, καθώς και ποιούς πόρους έχουν ήδη δεσμεύσει εκείνο το χρονικό διάστημα. [3]

KanseiGenie

Η ερευνητική ομάδα KanseiGenie έχει αναπτύξει μια πλατφόρμα για διαχείριση, εικονοποίηση και ενοποίηση δικτύων ασύρματων αισθητήρων (Wireless sensor networks (WSN)). Οι διάφοροι χρήστες προκειμένου να πραγματοποιήσουν τα πειράματα τους θα πρέπει να βρουν τους διαθέσιμους πόρους, να τους δεσμεύσουν και να καθορίσουν τα χαρακτηριστικά τους. Οι πόροι εκχωρούνται σε κάποιο slice, το οποίο περιλαμβάνει ένα ή περισσότερα slivers της πειραματικής πλατφόρμας, τα οποία είναι τα επιμέρους κομμάτια που προκύπτουν από την εικονοποίηση των ασύρματων αισθητήρων. Τα slices που δημιουργούνται μπορούν να ελέγχονται από έναν ή περισσότερους χρήστες, οι οποίοι έχουν οριστεί ως διαχειριστές του slice και μπορούν να το τροποποιήσουν και να το χρησιμοποιήσουν για τα πειράματα τους. Ως εκ τούτου στο κομμάτι της διαχείρισης των πόρων της πειραματικής πλατφόρμας η ερευνητική ομάδα έχει αναπτύξει ένα εργαλείο που περιλαμβάνει υπηρεσίες για δημοσίευση των διαθέσιμων πόρων (resource publishing), την εύρεση των κατάλληλων πόρων της πειραματικής πλατφόρμας (resource discovery), την δέσμευση των πόρων (resource reservation), καθώς και τον καθορισμό των χαρακτηριστικών των πόρων που έχουν δεσμευθεί (resource configuration) με σκοπό την πραγματοποίηση των διάφορων πειραμάτων (όπως αναφέρεται στο [14]). Ας εξετάσουμε αναλυτικά πως λειτουργεί κάθε μια από τις παραπάνω υπηρεσίες, όπως περιγράφεται στο [14]:

- **Resource publishing**

Κάθε δίκτυο ασύρματων αισθητήρων θα πρέπει να ανακοινώνει την πληροφορία για τη διαθεσιμότητα των πόρων που διαθέτει σε μια ή περισσότερες

Κεφάλαιο 1:Εισαγωγή

οντότητες που έχουν ονομαστεί clearinghouses. Τα clearinghouses περιέχουν πληροφορίες για τους πιστοποιημένους χρήστες, τους πόρους που είναι διαθέσιμοι από τα διάφορα δίκτυα ασύρματων αισθητήρων καθώς και για τις κρατήσεις που έχουν γίνει και για τα slices που έχουν δημιουργηθεί. Με τη δημοσίευση των διαθέσιμων πόρων σε ένα ή περισσότερα clearinghouses ένας μεγάλος αριθμός χρηστών που επικοινωνούν με αυτά μπορούν να έχουν πρόσβαση στους διαθέσιμους πόρους και να τους δεσμεύσουν για τα πειράματα τους. Ως εκ τούτου πετυχαίνουμε καλύτερη αξιοποίηση των διαθέσιμων πόρων των δικτύων ασύρματων αισθητήρων.

- Resource discovery

Η υπηρεσία αυτή έχει ως στόχο να εντοπίσει τους κατάλληλους πόρους της πειραματικής πλατφόρμας που θέλει ο ερευνητής για τα πειράματα του. Η υπηρεσία αυτή προϋποθέτει την ύπαρξη κατάλληλης γλώσσας περιγραφής των πόρων έτσι ώστε ο ερευνητής να μπορεί να περιγράψει με λεπτομέρεια τους πόρους που θέλει να δεσμεύσει για τα πειράματα του. Η υπηρεσία εύρεσης των κατάλληλων πόρων θα πρέπει να έχει τη δυνατότητα να ερμηνεύει τα μηνύματα των διάφορων χρηστών και να εντοπίζει τους κατάλληλους πόρους για να ικανοποιήσει τις απαιτήσεις τους.

- Resource reservation

Μόλις ολοκληρωθεί το παραπάνω στάδιο που επιστρέφει στο χρήστη τους κατάλληλους πόρους της πειραματικής πλατφόρμας ανάλογα με την περιγραφή που έχει δώσει θα πρέπει να πραγματοποιηθεί η δέσμευση των πόρων για κάποιο συγκεκριμένο χρονικό διάστημα. Η υπηρεσία αυτή θα πρέπει να επικοινωνήσει με τον πάροχο των πόρων για να ελέγξει τη διαθεσιμότητα τους εκείνο το χρονικό διάστημα που θέλει να τους χρησιμοποιήσει ο ερευνητής. Εφόσον είναι διαθέσιμοι δεσμεύονται από τον ερευνητή σε κάποιο slice του.

- Resource configuration

Οι πόροι που έχουν δεσμευθεί σε κάποιο slice του χρήστη θα πρέπει να ρυθμιστούν κατάλληλα ώστε να πραγματοποιηθούν τα πειράματα του χρήστη. Η υπηρεσία της ρύθμισης των πόρων είναι υπεύθυνη να παρέχει στο χρήστη όλες τις παραμέτρους που πρέπει να ρυθμιστούν για τη διεξαγωγή του πειράματος από το χρήστη. Οι παράμετροι αυτοί μπορεί να ποικίλουν από μια ρύθμιση στο λογισμικό του πόρου μέχρι καθορισμό της ισχύος εκπομπής του ασύρματου αισθητήρα και καθορισμός της ακριβούς τοπολογίας του δικτύου για το πείραμα.

Οι διάφοροι ερευνητές μπορούν να ζητήσουν μέσω κατάλληλης προγραμματιστικής διεπαφής (API) συγκεκριμένους πόρους για τα πειράματα τους. Εφόσον αυτοί οι πόροι είναι διαθέσιμοι δεσμεύονται σε ένα slice του χρήστη για να χρησιμοποιηθούν για τα πειράματα του. Έτσι με την ανάπτυξη αυτού του εργαλείου διαχείρισης από την ερευνητική ομάδα του KanseiGenie οι ερευνητές μπορούν να

Κεφάλαιο 1:Εισαγωγή

σχεδιάζουν και να διαχειρίζονται τα πειράματα τους δεσμεύοντας τους πόρους που επιθυμούν.

Κεφάλαιο 2

Βασικά Πρωτόκολλα και Τεχνολογίες

2.1 Η γλώσσα XML

2.1.1 Περιγραφή της γλώσσας XML

Η XML (eXtensible Markup Language) ([15],[16]) είναι μια γλώσσα σήμανσης για έγγραφα που περιέχουν δομημένα δεδομένα. Για την ακρίβεια είναι κάτι περισσότερο από γλώσσα σήμανσης, είναι μετα-γλώσσα, δηλαδή μια γλώσσα που χρησιμοποιείται για να καθορίσει νέες γλώσσες σήμανσης. Η XML μας παρέχει ένα ορισμένο τρόπο για δόμηση, περιγραφή και ανταλλαγή πληροφοριών.

Η XML προέκυψε από τον συνδυασμό δύο τεχνολογιών: της SGML και της HTML [17]. Η Πρότυπη Γενική Γλώσσα Σήμανσης (Standard Generalized Markup Language, SGML) είναι μια μετα-γλώσσα που επιτρέπει τον ορισμό γλωσσών ανταλλαγής δεδομένων και εγγράφων. Η XML αναπτύχθηκε για να διατηρήσει όλη την περιγραφική δύναμη της SGML αφαιρώντας όλη σχεδόν την πολυπλοκότητα της έτσι ώστε να είναι απλή στη χρήση της..

Το XML κείμενο έχει δομή δέντρου, η οποία είναι αρκετά απλή και αποτελείται από τα εξής μέρη:

1. Στοιχεία (elements)
2. Ετικέτες (tags)
3. Γνωρίσματα (attributes)
4. Οντότητες (entities)
5. Σχόλια (comments)

Η γλώσσα XML χρησιμοποιείται ευρέως σε διαδικτυακές εφαρμογές για τη μεταφορά πληροφοριών σε δομημένη μορφή. Επιπλέον χρησιμοποιείται στις διάφορες υπηρεσίες ιστού (Web Services), που έχουν ραγδαία ανάπτυξη τα τελευταία χρόνια στο διαδίκτυο, ως κύρια δομή οργάνωσης των πληροφοριών που ανταλλάσσονται. Για αυτό το λόγο έχουν αναπτυχθεί πολλές εφαρμογές ώστε οι χρήστες να μπορούν να επεξεργαστούν τις πληροφορίες του XML κειμένου (XML parsers).

Ένα κείμενο XML συνήθως περιλαμβάνει ένα σχήμα με το οποίο θα πρέπει να συμμορφώνεται η δομή του. Έχουν αναπτυχθεί διάφορα συστήματα προς αυτή την κατεύθυνση της εγκυρότητας (validation) του XML κειμένου. Δυο από αυτά τα οποία χρησιμοποιήσαμε στην παρούσα διπλωματική είναι το XML Schema (xsd) [18] και το RELAX NG [20] που θα περιγράψουμε σε επόμενα υποκεφάλαια. Αν τα διάφορα στοιχεία (elements) και χαρακτηριστικά (attributes) του XML κειμένου είναι

σύμφωνα με την δομή που περιγράφεται από το σχήμα τότε το XML κείμενο καλείται καλώς δομημένο (well-formed) και έγκυρο (valid).

2.1.2 Πλεονεκτήματα της XML

Η XML είναι ιδιαίτερα χρήσιμη σε αρκετές περιπτώσεις και ειδικά υπηρεσίες και εφαρμογές διαδικτύου. Μερικά από τα πλεονεκτήματα της XML είναι:

- **Χρήση σε ετερογενείς εφαρμογές**

Η XML είναι χρήσιμη όσον αφορά τη χρήση της σε ετερογενή περιβάλλοντα. Μπορεί να χρησιμοποιηθεί σε διάφορες πλατφόρμες γιατί έχουν αναπτυχθεί XML αναλυτές σύνταξης (parsers) σχεδόν για όλες τις πλατφόρμες και τις προγραμματιστικές γλώσσες. Ως συνέπεια η XML προτιμάται για τη μεταφορά δεδομένων σε ετερογενείς πλατφόρμες. Οι πλατφόρμες αυτές δεν είναι απαραίτητο να χρησιμοποιούν το ίδιο λειτουργικό σύστημα αφού θα μπορούν όλες να καταλάβουν και να επεξεργαστούν τα XML κείμενα που ανταλλάσσονται.

- **Εύκολη αναζήτηση πληροφοριών**

Η XML γλώσσα βοηθάει τους χρήστες στην εύκολη αναζήτηση των πληροφοριών που χρειάζονται. Με τη χρήση κατάλληλων XML αναλυτών σύνταξης (parsers) δίνεται η δυνατότητα να διασχίσουμε τη δενδρική δομή του XML κειμένου και να επιλέξουμε τα στοιχεία που μας ενδιαφέρουν. Ως εκ τούτου η αναζήτηση και εξαγωγή στοιχείων είναι σχετικά εύκολη όταν οι πληροφορίες ανταλλάσσονται σε XML δομή.

- **Έλεγχος εγκυρότητας (validation)**

Τα κείμενα XML μπορούν να ελεγχθούν για την εγκυρότητα τους με βάση κάποια σχήματα. Όπως αναφέραμε προηγουμένως έχουν αναπτυχθούν αρκετά εργαλεία προς αυτή την κατεύθυνση και έτσι είναι απλό να γίνει ο έλεγχος της εγκυρότητας του XML κειμένου.

Ορίζοντας ένα σύνδεσμο για το σχήμα με το οποίο πρέπει να συμμορφώνεται η XML δομή γίνεται ο έλεγχος της ορθότητας του XML. Στην παρούσα διπλωματική αυτό είναι πολύ σημαντικό γιατί θέλουμε να ελέγξουμε αν η περιγραφή των χαρακτηριστικών ασύρματων κόμβων του εργαστηρίου είναι σύμφωνη με κάποια πρότυπα. Η περιγραφή είναι ένα XML κείμενο το οποίο πρέπει να περιέχει χαρακτηριστικά του κόμβου με μια συγκεκριμένη δομή. Έτσι αν λάβουμε περιγραφή μη έγκυρη με ένα σχήμα θα πρέπει να την απορρίψουμε και να μην συνεχίσουμε την περαιτέρω επεξεργασία των πληροφοριών που περιέχει.

2.2 XML Σχήμα (xsd)

Το XML Σχήμα (XML Schema (xsd)) [18] είναι ένα έγγραφο βασισμένο στην XML προδιαγραφή για την περιγραφή της δομής που πρέπει να ακολουθείται από ένα XML κείμενο. Ο όρος Σχήμα έχει την έννοια της ύπαρξης ενός αρχείου διάταξης. Ένα XML σχήμα:

1. Ορίζει τα στοιχεία (elements) σε ένα XML κείμενο
2. Ορίζει τις ιδιότητες (attributes) σε ένα XML κείμενο
3. Ορίζει ποία στοιχεία είναι στοιχεία-παιδιά
4. Ορίζει την ακολουθία των στοιχείων-παιδιών
5. Ορίζει αν ένα στοιχείο είναι άδειο ή περιέχει κάποιο κείμενο
6. Ορίζει τους τύπους δεδομένων των στοιχείων και των ιδιοτήτων στο XML κείμενο.
7. Ορίζει προκαθορισμένες τιμές για στοιχεία και ιδιότητες.

Συνεπώς το XML Σχήμα συνεισφέρει ώστε το XML κείμενο να έχει μια καλώς δομημένη μορφή σύμφωνα με κάποια πρότυπα. Ένα απλό παράδειγμα, που πήραμε από το [21], παρατίθεται παρακάτω:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="note">
<xs:complexType>
<xs:sequence>
<xs:element name="to" type="xs:string"/>
<xs:element name="from" type="xs:string"/>
<xs:element name="heading" type="xs:string"/>
<xs:element name="body" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Πίνακας 2.1: Παράδειγμα XML σχήματος [21]

Στο παραπάνω παράδειγμα ορίζεται ότι το XML κείμενο έχει ένα κύριο στοιχείο-ρίζα που θα είναι το note. Το στοιχείο note είναι σύνθετου τύπου (complex type) και περιέχει μια ακολουθία έως και τεσσάρων στοιχείων, τα οποία καλούνται παιδιά του αρχικού στοιχείου. Τα παιδιά του στοιχείου είναι τα στοιχεία to, from, heading και body. Τα στοιχεία αυτά είναι τύπου string για να είναι έγκυρο το XML κείμενο που βασίζεται σε αυτό το σχήμα. Αντίστοιχα θα μπορούσε τα παιδιά να είναι είτε σύνθετου τύπου (complex type) είτε απλού τύπου (simple type) είτε ένας πίνακας στοιχείων κάποιου τύπου (arrayType).

Επιπλέον στο XML Σχήμα δίνεται η δυνατότητα να χρησιμοποιήσουμε τύπους στοιχείων που έχουν οριστεί σε κάποιο άλλο σχήμα αρκεί να συμπεριλάβουμε το σχήμα αυτό στο σχήμα μας. Με αυτό τον τρόπο μπορούμε να χρησιμοποιήσουμε οποιοδήποτε τύπο δεδομένων έχει οριστεί στο άλλο XML Σχήμα χωρίς να χρειάζεται να ξαναορίσουμε τους διάφορους τύπους στο καινούργιο σχήμα. Αυτό γίνεται κάνοντας include το σχήμα στο δικό μας.

Κεφάλαιο 2: Βασικά Πρωτόκολλα και Τεχνολογίες

Επομένως φτιάχνοντας το XML Σχήμα μας με την κατάλληλη δομή και τους κατάλληλους τύπους στοιχείων που πρέπει να έχει το XML κείμενο μας μπορούμε να ελέγξουμε αν είναι έγκυρο και στη συνέχεια να επεξεργαστούμε τις πληροφορίες του. Έχουν αναπτυχθεί πολλά εργαλεία τα οποία ελέγχουν αν είναι έγκυρο το XML κείμενο με βάση ένα ή περισσότερα σχήματα σε σχεδόν όλες τις γλώσσες προγραμματισμού. Το μειονέκτημα του XML Σχήμα είναι ότι είναι πολύπλοκο να το συντάξει και δύσκολο να το διαχειριστεί ο χρήστης.

2.3 Πεδίο Ονομάτων (*namespace*)

Ένα πεδίο ονομάτων (*namespace*) [19] είναι μια συλλογή από ονόματα αναγνωρισμένα από μια αναφορά, τα οποία χρησιμοποιούνται στα XML κείμενα σαν τύποι στοιχείων και ονόματα ιδιοτήτων. Παρέχουν μια απλή μέθοδο περιγραφής στοιχείων και ονομάτων ιδιοτήτων που χρησιμοποιούνται στα XML κείμενα και τα συσχετίζουν με τα namespaces που αναγνωρίζονται από τις URI (Uniform Resource Identifier) αναφορές. Με τα namespaces εξασφαλίζεται ανεξαρτησία στην επιλογή του κειμένου των ετικετών, δηλαδή στην ονομασία των επιμέρους στοιχείων ανάμεσα σε διαφορετικά κείμενα.

Ένα παράδειγμα δήλωσης namespace από το [19], το οποίο συσχετίζει το prefix namespace *edi* με το όνομα του namespace `http://ecommerce.example.org/schema` είναι το παρακάτω:

```
<x xmlns:edi='http://ecommerce.example.org/schema'>
  <!-- the "edi" prefix is bound to http://ecommerce.example.org/schema
        for the "x" element and contents -->
</x>
```

Πίνακας 2.2: Παράδειγμα namespace [19]

2.4 Relax NG

Η Relax NG [20] είναι μια γλώσσα για την περιγραφή της δομής ενός XML κειμένου καθώς και των τύπων των διάφορων στοιχείων που το απαρτίζουν. Ένα RELAX NG σχήμα καθορίζει ένα πρότυπο για την δομή και το περιεχόμενο ενός XML κειμένου. Συνεπώς χρησιμοποιώντας ένα Relax NG σχήμα μπορούμε να ελέγξουμε ως προς την εγκυρότητα ένα XML κείμενο.

Η γλώσσα αυτή δεν είναι τόσο πολύπλοκη όσο είναι η XML Schema και είναι πιο εύκολο στο χρήστη να σχεδιάσει το σχήμα του χρησιμοποιώντας τη *rnc* περιγραφή. Αυτό είναι και το μεγάλο πλεονέκτημα της σε σύγκριση με την XML Schema. Η Compact σύνταξη (Syntax) της Relax NG ((Relax NG Compact Syntax(*rnc*)) χρησιμοποιεί κάποια σύμβολα για να δηλώσει τι τύπου είναι τα στοιχεία και το πόσες φορές μπορούν να εμφανίζονται μέσα στο XML κείμενο. Τα παιδιά που έχει κάποιο στοιχείο εμπεριέχονται μέσα σε αγκύλες, όπως επίσης και ο τύπος του στοιχείου αν είναι απλού τύπου. Επιπλέον υπάρχουν κάποια σύμβολα για να δηλώσουν πόσες φορές μπορεί να εμφανιστεί το στοιχείο στη δομή XML. Το σύμβολο * δηλώνει

Κεφάλαιο 2: Βασικά Πρωτόκολλα και Τεχνολογίες

μηδέν ή περισσότερες φορές, το σύμβολο + δηλώνει μία τουλάχιστον φορά και το σύμβολο ? υποδηλώνει ότι το στοιχείο είναι προαιρετικό.

Ας δούμε ένα απλό παράδειγμα rnc σχήματος από το [20]:

```
element addressBook {
  element card {
    element name { text },
    element email { text },
    element note { text }?
  }*
}
```

Πίνακας 2.3: Παράδειγμα RelaxNG Compact Syntax [20]

Στο παραπάνω παράδειγμα βλέπουμε ότι το XML κείμενο θα αποτελείται από ένα αρχικό στοιχείο addressBook. Το στοιχείο αυτό θα έχει μηδέν ή περισσότερα στοιχεία «παιδιά» card, όπως υποδηλώνει ο τελεστής * του rnc. Το στοιχείο card θα έχει τρία παιδιά τα name, email και note, εκ των οποίων το στοιχείο note είναι προαιρετικό. Ως εκ τούτου το XML κείμενο θα πρέπει να συμμορφώνεται με βάση αυτούς τους περιορισμούς για να είναι έγκυρο (validate).

Συνεπώς το Relax NG μας παρέχει ένα εύχρηστο εργαλείο για την κατασκευή των διάφορων σχημάτων μας, με τα οποία πρέπει να συμμορφώνονται τα XML κείμενα. Αξίζει να επισημάνουμε ότι έχουν αναπτυχθεί διάφορα εργαλεία για να μετατρέψουν το φιλικό στο χρήστη rnc αρχείο σε XML Schema (xsd) αρχείο το οποίο υποστηρίζεται από τα περισσότερα προγράμματα που ελέγχουν την εγκυρότητα των XML. Ένα τέτοιο εργαλείο είναι το trang, το οποίο χρησιμοποιήσαμε στην παρούσα διπλωματική για να μετατρέψουμε rnc αρχείο σε xsd. Καταλαβαίνουμε λοιπόν ότι με τη χρήση της Relax NG γλώσσας μας δίνεται η δυνατότητα να σχεδιάσουμε εύκολα τα διάφορα σχήματα μας.

2.5 Namespace-based Validation Dispatching Language (NVDL)

Η Namespace-based Validation Dispatching Language (NVDL) [22] είναι μια γλώσσα βασισμένη στην XML προδιαγραφή που χρησιμοποιείται για τον έλεγχο της εγκυρότητας XML αρχείων που χρησιμοποιούν περισσότερα από ένα πεδίο ονομάτων.

Τα περισσότερα XML κείμενα χρησιμοποιούν ένα μοναδικό namespace. Σε αυτή την περίπτωση είναι σχετικά απλός ο έλεγχος της εγκυρότητας του XML κειμένου με βάση εργαλεία που έχουν αναπτυχθεί σε διάφορες γλώσσες προγραμματισμού. Το πρόβλημα είναι όταν το XML κείμενο χρησιμοποιεί στοιχεία που ανήκουν σε περισσότερα από ένα namespaces και τότε ο έλεγχος εγκυρότητας γίνεται περίπλοκος. Με τη χρήση της NVDL γλώσσας μπορούμε να ελέγξουμε XML αρχεία που έχουν περισσότερα από ένα namespaces.

Τα NVDL αρχεία, όπως αναφέρεται στο [23], περιέχουν ένα σύνολο από κανόνες (rules), ο καθένας από τους οποίους πραγματοποιεί κάποιες ενέργειες

Κεφάλαιο 2: Βασικά Πρωτόκολλα και Τεχνολογίες

(actions). Οι ενέργειες αυτές μπορεί να περιλαμβάνουν τον έλεγχο της εγκυρότητας (validate) κάποιων στοιχείων με δεδομένο σχήμα όταν αυτά ανήκουν σε ένα συγκεκριμένο namespace. Για να πραγματοποιήσουμε αυτή την ενέργεια πρέπει να ορίσουμε το namespace και στη συνέχεια να δηλώσουμε ότι θα το κάνουμε validate με κάποιο σχήμα (schema), το οποίο θα διευκρινίζεται από ένα URL.

Ας δώσουμε ένα παράδειγμα ενός nvdI αρχείο που χρησιμοποιήσαμε στην παρούσα διπλωματική για τον έλεγχο της εγκυρότητας ενός XML αρχείου που περιείχε δύο namespaces:

```
<rules xmlns="http://purl.oclc.org/dsdl/nvdI/ns/structure/1.0">

  <namespace ns="http://sorch.netmode.ntua.gr/ws/RSpec">
    <validate schema="http://sorch.netmode.ntua.gr/ws/RSpec/request.xsd"/>
  </namespace>
  <namespace ns="http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if">
    <validate schema=
      "http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if/interfaces.xsd"/>
  </namespace>

</rules>
```

Πίνακας 2.4: Παράδειγμα NVDL αρχείου

Στο παραπάνω παράδειγμα βλέπουμε ότι αρχικά ορίζουμε το namespace των επιμέρους στοιχείων και στην συνέχεια δηλώνουμε το action το οποίο θέλουμε να πραγματοποιήσουμε, το οποίο είναι το validate, δηλαδή ο έλεγχος της εγκυρότητας. Στη συνέχεια ορίζουμε το attribute schema στο οποίο εισάγουμε το URL του σχήματος με το οποίο θέλουμε να κάνουμε validation των αντίστοιχων στοιχείων που αναφέρονται στο συγκεκριμένο namespace. Έτσι για τα στοιχεία με namespace `http://sorch.netmode.ntua.gr/ws/RSpec` θέλουμε να κάνουμε validate με το xsd σχήμα που βρίσκεται στο URL `http://sorch.netmode.ntua.gr/ws/RSpec/request.xsd`. Ομοίως για τα στοιχεία με namespace `http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if` θέλουμε να κάνουμε validate με το xsd σχήμα που βρίσκεται στο URL `http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if/interfaces.xsd`.

Συνεπώς με τη χρήση της NVDL γλώσσας μπορούμε να κάνουμε validation XML κειμένων που περιέχουν περισσότερα από ένα namespaces, επιλέγοντας τα στοιχεία που ανήκουν στο συγκεκριμένο namespace και εφαρμόζοντας του έλεγχο εγκυρότητας με το κατάλληλο σχήμα. Αυτό ήταν κάτι που βοήθησε σημαντικά στην υλοποίηση της παρούσας διπλωματικής αφού στα RSpec αρχεία που θα αναφέρουμε αναλυτικά σε επόμενο κεφάλαιο υπήρχαν περισσότερα του ενός namespaces τα στοιχεία των οποίων έπρεπε να ελεγχθούν ως προς την εγκυρότητα με βάση κάποια συγκεκριμένα data models.

2.6 Ενιαίο Όνομα Πόρου

Το ενιαίο όνομα πόρου (Uniform Resource Name-URN) αποτελεί το αναγνωριστικό (identifier) των διάφορων πόρων της πειραματικής πλατφόρμας

Κεφάλαιο 2: Βασικά Πρωτόκολλα και Τεχνολογίες

(testbed), όπως αναφέρεται στο [24]. Τα URNs χρησιμοποιούνται για την αναγνώριση των διάφορων πόρων ανάλογα με τον τύπο και το όνομα τους. Χρησιμοποιούνται ως αναγνωριστικά σε πολλές πειραματικές πλατφόρμες όπως για παράδειγμα στις πειραματικές πλατφόρμες του GENI [25].

Το μεγάλο πλεονέκτημα των URNs ως identifiers είναι ότι έχουν μια καθορισμένη δομή σύμφωνα με την πρόταση του GMOC. Συγκεκριμένα αυτή η δομή, όπως περιγράφεται στο [24], περιλαμβάνει toplevelauthority, τον τύπο του πόρου (resource) και το όνομα του και συνεπώς έχει την εξής μορφή:

urn:publicid:IDN+toplevelauthority+resource-type+resource-name

Ας αναλύσουμε τώρα τα επιμέρους στοιχεία του URN. Το toplevelauthority είναι ένα διεθνώς αναγνωρίσιμο όνομα domain στο οποίο ανήκει ο συγκεκριμένος πόρος. Ο τύπος του πόρου μπορεί να είναι:

1. node: ο πόρος είναι ένας κόμβος της πειραματικής πλατφόρμας.
2. interface: μια διεπαφή της πειραματικής πλατφόρμας.
3. link: μια ζεύξη της πειραματικής πλατφόρμας.
4. slice: ένα slice που περιέχει πόρους της πειραματικής πλατφόρμας.
5. sliver: κομμάτι ενός κόμβου που ανήκει σε κάποιο slice.
6. authority: μια υπηρεσία της πειραματικής πλατφόρμας.

Επιπλέον το όνομα του resource θα πρέπει να είναι μοναδικό ανάμεσα στα resources που ανήκουν στην ίδια πειραματική πλατφόρμα έτσι ώστε να αναγνωρίζεται κατάλληλα.

Παρέχοντας αυτές τις πληροφορίες μπορούμε να αναγνωρίσουμε κατάλληλα το resource ότι ανήκει σε κάποια πειραματική πλατφόρμα και επιπλέον μπορούμε να εξαγάγουμε εύκολα το όνομα του resource για να πάρουμε τις πληροφορίες που θέλουμε για αυτό.

Στην παρούσα διπλωματική χρησιμοποιήσαμε τα URNs ως identifiers των πόρων στο RSpec. Συγκεκριμένα χρησιμοποιήσαμε URNs για τρεις διαφορετικούς τύπους πόρων.

Ο πρώτος τύπος είναι κόμβος (node) της πειραματικής πλατφόρμας και χρησιμοποιήσαμε URN **urn:publicid:IDN+sorch.netmode.ntua.gr+node+omf.netmode.node1**. Το URN αυτό περιλαμβάνει ότι το toplevelauthority της πειραματικής πλατφόρμας είναι sorch.netmode.ntua.gr, ο τύπος του resource που περιγράφεται είναι κόμβος (node) και το όνομα του είναι omf.netmode.node1, το οποίο είναι μοναδικό μεταξύ των κόμβων της πειραματικής πλατφόρμας. Ο δεύτερος τύπος είναι μια διεπαφή του κόμβου (interface) και για την περιγραφή της χρησιμοποιήσαμε το URN **urn:publicid:IDN+sorch.netmode.ntua.gr+interface+omf.netmode.node1:if0**. Το URN αυτό περιλαμβάνει ότι το toplevelauthority της πειραματικής πλατφόρμας είναι sorch.netmode.ntua.gr, ο τύπος του resource που περιγράφεται είναι διεπαφή (interface) και το όνομα του είναι omf.

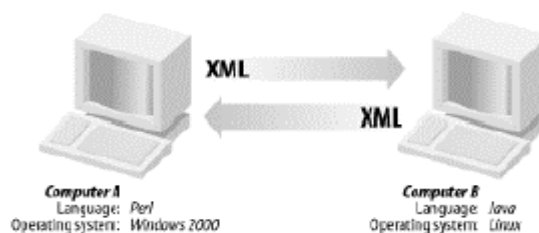
netmode.nodel:if0, δηλαδή το 1^ο interface του κόμβου 1, το οποίο είναι μοναδικό μεταξύ των διεπαφών της πειραματικής πλατφόρμας. Ο τρίτος τύπος είναι μια υπηρεσία της πειραματικής πλατφόρμας (authority) και χρησιμοποιήσαμε το URN **urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm**. Το URN αυτό περιγράφει τον component manager (cm) των διαφόρων resources της εργαστηριακής πλατφόρμας και περιλαμβάνει το toplevelauthority που είναι το sorch.netmode.ntua.gr, τον τύπο του resource που είναι μια υπηρεσία (authority) και το όνομα του authority που είναι cm, δηλαδή component manager.

2.7 Web Services

2.7.1 Περιγραφή των υπηρεσιών διαδικτύου (Web services)

Μια υπηρεσία διαδικτύου (web service) σύμφωνα με το World Wide Web Consortium είναι ότι αποτελεί μια εφαρμογή λογισμικού που αναγνωρίζεται από ένα URI, της οποίας το interface και τα bindings μπορούν να αναγνωριστούν, περιγραφούν και ανακαλυφθούν με τη βοήθεια της γλώσσας XML και επιπλέον υποστηρίζει απ' ευθείας αλληλεπιδράσεις με άλλες λογισμικές εφαρμογές χρησιμοποιώντας μηνύματα βασισμένα σε XML μέσω πρωτοκόλλων βασισμένων στο Internet, όπως για παράδειγμα το SOAP πρωτόκολλο που θα αναλύσουμε σε επόμενο υποκεφάλαιο. [26]

Με πιο απλά λόγια μια web service είναι μια υπηρεσία διαθέσιμη στο διαδίκτυο, η οποία χρησιμοποιεί ένα καθορισμένο σύστημα μηνυμάτων βασισμένο σε XML και δεν είναι δεμένη με κανένα λειτουργικό σύστημα ή γλώσσα προγραμματισμού. Παραθέτουμε παρακάτω μια εικόνα που δείχνει ότι η υπηρεσία διαδικτύου είναι ανεξάρτητη από λειτουργικό σύστημα και γλώσσα προγραμματισμού. Συγκεκριμένα βλέπουμε ότι ο ένας υπολογιστής χρησιμοποιεί λειτουργικό σύστημα windows και γλώσσα προγραμματισμού perl, ενώ ο άλλος λειτουργικό σύστημα Linux και γλώσσα προγραμματισμού java, όμως μπορούν και οι δύο να ανταλλάξουν πληροφορίες σε XML μορφή μέσω υπηρεσιών διαδικτύου.



Σχήμα 2.1: Υπηρεσία ιστού ανεξάρτητη από λειτουργικό σύστημα και γλώσσα προγραμματισμού

Κεφάλαιο 2: Βασικά Πρωτόκολλα και Τεχνολογίες

Συνοπτικά μια ολοκληρωμένη web service είναι μια υπηρεσία η οποία:

1. Είναι διαθέσιμη στο διαδίκτυο ή σε ιδιωτικά (intranet) δίκτυα.
2. Χρησιμοποιεί ένα καθορισμένο σύστημα μηνυμάτων βασισμένο σε XML.
3. Δεν είναι δεμένη με κανένα λειτουργικό σύστημα ή γλώσσα προγραμματισμού.
4. Είναι αυτό-περιγραφόμενη μέσω κοινής XML γραμματικής.
5. Είναι ανακαλύψιμη μέσω μιας απλής διαδικασίας εύρεσης.

2.7.2 Γιατί να χρησιμοποιήσουμε *Web Services*

Παραδοσιακά οι web εφαρμογές παρέχουν την δυνατότητα αλληλεπίδρασης μεταξύ του τελικού χρήστη και της δικτυακής σελίδας ενώ οι web services είναι service-oriented και παρέχουν τη δυνατότητα επικοινωνίας από εφαρμογή σε εφαρμογή μέσα στο διαδίκτυο και εύκολη πρόσβαση σε ετερογενείς εφαρμογές και συσκευές. Τα παρακάτω είναι οι βασικότεροι τεχνικοί λόγοι για να επιλέξετε τις web services αντί των web εφαρμογών:

- Οι web services μπορούν να κληθούν με τη βοήθεια RPC μηχανισμών βασισμένων στην XML χωρίς να εμποδίζονται από firewalls.
- Οι web services παρέχουν μια λύση που εφαρμόζεται σε όλες τις πλατφόρμες και γλώσσες και βασίζεται στην ανταλλαγή XML μηνυμάτων.
- Οι web services διευκολύνουν την υλοποίηση των εφαρμογών χρησιμοποιώντας μια υποδομή που δεν επηρεάζει τη δυνατότητα κλιμάκωσης.
- Οι web services καθιστούν δυνατή τη διαλειτουργικότητα μεταξύ ετερογενών εφαρμογών.

2.7.3 Βασικά πρωτόκολλα για υπηρεσίες διαδικτύου (*Web Services*)

Οι υπηρεσίες διαδικτύου βασίζονται σε κάποια πρωτόκολλα για τη μεταφορά των δεδομένων σε XML μορφή καθώς και για την περιγραφή των υπηρεσιών διαδικτύου και των λειτουργιών τους. Για τη μεταφορά των μηνυμάτων που ανταλλάσσονται χρησιμοποιείται το πρωτόκολλο SOAP [27] και για την περιγραφή των υπηρεσιών διαδικτύου χρησιμοποιείται η WSDL γλώσσα [28].

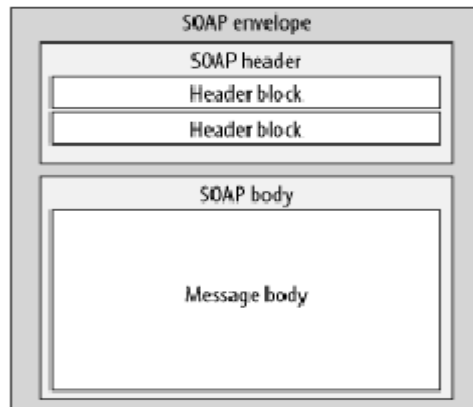
2.7.3.1 SOAP (Simple Object Access Protocol) – Πρωτόκολλο Μεταφοράς

Το SOAP [27] είναι ένα τυποποιημένο πρωτόκολλο που «πακετάρει» τα μηνύματα τα οποία μοιράζονται και ανταλλάσσουν οι διάφορες εφαρμογές. Η προδιαγραφή του SOAP δεν καθορίζει τίποτα άλλο παρά έναν απλό φάκελο βασισμένο στην XML για να μεταφέρει τις διάφορες πληροφορίες μαζί με ένα σύνολο κανόνων για μετάφραση και μετατροπή των διαφόρων τύπων δεδομένων σε XML εκφράσεις. Το SOAP παρέχει ένα πρότυπο κωδικοποιημένο τρόπο για την κατασκευή XML μηνυμάτων, ο οποίος αναγνωρίζεται από ετερογενείς εφαρμογές που ανταλλάσσουν πληροφορίες μέσω των XML μηνυμάτων. Ως εκ τούτου ο

Κεφάλαιο 2: Βασικά Πρωτόκολλα και Τεχνολογίες

σχεδιασμός του SOAP το κάνει κατάλληλο για χρήση από μια μεγάλη ποικιλία εφαρμογών που χρησιμοποιούν μηνύματα για επικοινωνία και από πολλά ολοκληρωμένα μοντέλα, ανεξαρτήτως λειτουργικού συστήματος, γλώσσας προγραμματισμού ή άλλης τεχνικής λεπτομέρειας υλοποίησης που χρησιμοποιούν.

Το XML μήνυμα που δημιουργείται με βάση το SOAP ονομάζεται μήνυμα SOAP. Ένα SOAP μήνυμα, όπως αναφέρεται στο [27], αποτελείται από έναν φάκελο (SOAP envelope) που περιλαμβάνει μια προαιρετική επικεφαλίδα (SOAP header) και ένα υποχρεωτικό σώμα (SOAP body), όπως φαίνεται και στο παρακάτω σχήμα:



Σχήμα 2.2: SOAP envelope

Η επικεφαλίδα περιέχει πληροφορίες σχετικές με το πώς θα πρέπει να επεξεργαστεί το μήνυμα. Οι πληροφορίες αυτές αφορούν ρυθμίσεις δρομολόγησης και παραλαβής του μηνύματος, πιστοποιήσεις γνησιότητας και εξουσιοδοτήσεων και άλλες σχετικές με την ανταλλαγή των μηνυμάτων πληροφορίες.

Το σώμα περιέχει το πραγματικό μήνυμα το οποίο θα σταλεί και θα επεξεργαστεί. Οτιδήποτε το οποίο μπορεί να εκφραστεί με XML μπορεί και να σταλεί μέσα στο σώμα ενός SOAP μηνύματος.

Παραθέτουμε ένα παράδειγμα SOAP μηνύματος, από το [21] :

```
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Body xmlns:m="http://www.example.org/stock">
<m:GetStockPriceResponse>
<m:Price>34.5</m:Price>
</m:GetStockPriceResponse>
</soap:Body>
</soap:Envelope>
```

Πίνακας 2.5: Παράδειγμα SOAP envelope [21]

Στο παράδειγμα αυτό δεν υπάρχει η επικεφαλίδα, η οποία όπως είπαμε χρησιμοποιείται προαιρετικά στο SOAP φάκελο. Στο παραπάνω μήνυμα υπάρχουν:

Κεφάλαιο 2: Βασικά Πρωτόκολλα και Τεχνολογίες

- Το στοιχείο soap:Envelope, που είναι το εξωτερικό στοιχείο, το οποίο εμπεριέχει όλο το SOAP μήνυμα.
- Το στοιχείο soap:Body που ουσιαστικά περιέχει το μήνυμα που θα επεξεργαστεί. Το μήνυμα βλέπουμε ότι είναι η απάντηση (Response) στην κλήση της μεθόδου GetStockPrice στον server. Είναι ένα καλώς δομημένο XML μήνυμα του οποίου τις πληροφορίες θα επεξεργαστούμε. Συγκεκριμένα στο παράδειγμα μας μας επιστρέφεται η τιμή του προϊόντος, ως τιμή του στοιχείου Price.

2.7.3.2 WSDL (Web Service Description Language)

Η WSDL [28] είναι μια γλώσσα σε XML μορφή, η οποία χρησιμοποιείται για την περιγραφή των διάφορων υπηρεσιών διαδικτύου και των λειτουργιών τους. Η WSDL περιγράφει 4 σημαντικά κομμάτια δεδομένων:

- Πληροφορία για όλες τις διαθέσιμες λειτουργίες (operations).
- Πληροφορίες τύπων δεδομένων για όλα τα μηνύματα αίτησης και απάντησης (types).
- Πληροφορίες σύνδεσης σχετικά με τα πρωτόκολλα μεταφοράς (binding).
- Πληροφορίες διευθύνσεων URL για τον εντοπισμό της συγκεκριμένης υπηρεσίας.

Χρησιμοποιώντας την WSDL ένας χρήστης μπορεί να εντοπίσει μια web service και να καλέσει οποιαδήποτε από τις διαθέσιμες λειτουργίες της. Η δομή του WSDL κειμένου, όπως περιγράφεται στο [28], είναι η παρακάτω:

```
<definitions>
<types>
  <schema>
    definition of types.....
  </schema>
</types>
<message>
  <part>
    definition of a message....
  </part>
</message>
<portType>
  definition of a port.....
  <operation>
    <input></input>
    <output></output>
  </operation>
</portType>
<binding>
  definition of a binding....
  <operation>
    <input></input>
    <output></output>
  </operation>
</binding>
<service>
  definition of a service.....
```

```
<port></port>
</service>
</definitions>
```

Πίνακας 2.6: Δομή WSDL αρχείου [28]

Το WSDL αρχείο χωρίζεται σε 6 βασικά στοιχεία:

- **Definitions**

Το στοιχείο definitions πρέπει να είναι η ρίζα του δένδρου για κάθε WSDL αρχείο. Καθορίζει το όνομα της υπηρεσίας, δηλώνει τα πολλαπλά namespaces που χρησιμοποιούνται στο υπόλοιπο κείμενο και περιέχει όλα τα υπόλοιπα στοιχεία της υπηρεσίας.

- **Types**

Το στοιχείο types περιγράφει όλους τους τύπους δεδομένων που ανταλλάσσονται μεταξύ του πελάτη και του εξυπηρετητή. Η WSDL δεν ακολουθεί αποκλειστικά ένα συγκεκριμένο σύστημα δήλωσης τύπων, αλλά χρησιμοποιεί την W3C XML προδιαγραφή σχήματος ως προκαθορισμένη επιλογή της. Αν η υπηρεσία χρησιμοποιεί μόνο XML σχήμα δομημένο σε απλούς τύπους, όπως αλφαριθμητικά και ακεραίους, το στοιχείο types δεν είναι απαραίτητο.

- **Message**

Το στοιχείο message περιγράφει ένα μήνυμα μιας κατεύθυνσης είτε αυτό είναι ένα μήνυμα αίτησης είτε απάντησης για κάποια συγκεκριμένη λειτουργία της υπηρεσίας. Καθορίζει το όνομα του μηνύματος και περιέχει μηδέν ή περισσότερα στοιχεία ονομαζόμενα ως “part”, τα οποία αναφέρονται στις παραμέτρους του μηνύματος ή στις επιστρεφόμενες τιμές του.

- **PortType**

Το στοιχείο PortType συνδυάζει πολλαπλά στοιχεία ‘message’ για να συνθέσει μια απλή λειτουργία (operation) είτε μιας ή αμφίδρομης κατεύθυνσης. Για παράδειγμα, ένα PortType μπορεί να συνδυάζει ένα μήνυμα αίτησης και απάντησης σε μια μοναδική λειτουργία αίτησης-απάντησης η οποία χρησιμοποιείται πιο συχνά στις SOAP υπηρεσίες.

- **Binding**

Το στοιχείο binding περιγράφει το συγκεκριμένο τρόπο υλοποίησης της υπηρεσίας πάνω στο καλώδιο.

- **Service**

Το στοιχείο service καθορίζει τη διεύθυνση για την κλήση της συγκεκριμένης υπηρεσίας και περιέχει το URL της web service.

Παράδειγμα wsdl αρχείου και περιγραφή των στοιχείων του

Παραθέτουμε ένα αρχείο wsdl που χρησιμοποιήθηκε στην παρούσα διπλωματική για να κατανοήσουμε τις παραπάνω έννοιες. Το παρακάτω αρχείο περιγράφει την υπηρεσία mksliceservice.

```
<?xml version='1.0' encoding='UTF-8' ?>
<definitions name='RSpecSlice'
targetNamespace='http://sorch.netmode.ntua.gr/ws'
xmlns:tns='http://sorch.netmode.ntua.gr/ws'
xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'
xmlns:xsd1='http://sorch.netmode.ntua.gr/ws/'
xmlns='http://schemas.xmlsoap.org/wsdl/'>

<types>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://sorch.netmode.ntua.gr/ws"
targetNamespace="http://sorch.netmode.ntua.gr/ws"
xmlns:xsd1="http://sorch.netmode.ntua.gr/ws/">

<complexType name="RSpec">
<sequence>
<element name="name" type="xsd:string"/>
<element name="owner" type="xsd:string"/>
<element name="timestamp" type="xsd:string"/>
<element name="duration" type="xsd:int"/>
<element name="rspec1" type="xsd:string"/>
</sequence>
</complexType>
</schema>
</types>

<message name='mksliceRequest'>
<part name='Result' type='xsd1:RSpec'/>
</message>

<message name='mksliceResponse'>
<part name='Result' type='xsd:string'/>
</message>

<message name='modifysliceRequest'>
<part name='Result' type='xsd1:RSpec'/>
</message>

<message name='modifysliceResponse'>
<part name='Result' type='xsd:string'/>
</message>

<portType name='sliceoperationsPortType'>
<operation name='mkslice'>
<input message='tns:mksliceRequest'/>
<output message='tns:mksliceResponse'/>
</operation>

<operation name='modifyslice'>
<input message='tns:modifysliceRequest'/>
<output message='tns:modifysliceResponse'/>
</operation>
</portType>

<binding name='sliceoperationsBinding' type='tns:sliceoperationsPortType'>
<soap:binding style='rpc' transport='http://schemas.xmlsoap.org/soap/http/'>
```

Κεφάλαιο 2: Βασικά Πρωτόκολλα και Τεχνολογίες

```
<operation name='mkslice'>
<soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/mkslice'/>
<input>
<soap:body use='encoded' namespace='http://sorch.netmode.ntua.gr/ws'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
</input>
<output>
<soap:body use='encoded' namespace='http://sorch.netmode.ntua.gr/ws'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
</output>
</operation>

<operation name='modifyfyslice'>
<soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/modifyfyslice'/>
<input>
<soap:body use='encoded' namespace='http://sorch.netmode.ntua.gr/ws'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
</input>
<output>
<soap:body use='encoded' namespace='http://sorch.netmode.ntua.gr/ws'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
</output>
</operation>

</binding>

<service name='mksliceservice'>
<port name='sliceoperationsPortType' binding='sliceoperationsBinding'>
<soap:address location='http://sorch.netmode.ntua.gr/ws/mksliceservice.php'/>
</port>
</service>

</definitions>
```

Η υπηρεσία όπως φαίνεται από το WSDL αρχείο έχει δύο λειτουργίες (operations). Η πρώτη λειτουργία είναι η mkslice για τη δημιουργία ενός slice από κόμβους. Αυτή παίρνει ως όρισμα μια παράμετρο σύνθετου τύπου RSpec την οποία ορίσαμε στο στοιχείο types του WSDL αρχείου και επιστρέφει μια παράμετρο τύπου string. Η δεύτερη λειτουργία είναι η modifyfyslice για την τροποποίηση ενός slice. Αυτή η λειτουργία παίρνει ως όρισμα μια παράμετρο σύνθετου τύπου RSpec και επιστρέφει μια παράμετρο τύπου string. Ας εξετάσουμε όμως τα επιμέρους στοιχεία του παραδείγματος:

- **definitions**

Στο στοιχείο αυτό περιλαμβάνονται τα διάφορα namespaces που θα χρησιμοποιηθούν στο WSDL αρχείο. Η χρήση των namespaces είναι σημαντική για να ξεχωρίζουμε τα στοιχεία μεταξύ τους και δίνει την δυνατότητα στο WSDL κείμενο να αναφέρεται σε εξωτερικές προδιαγραφές, συμπεριλαμβανομένων και των WSDL, SOAP και XML Schema προδιαγραφών. Επιπλέον μπορούμε να ορίσουμε ένα δικό μας namespace για να ξεχωρίσουμε στοιχεία που ανήκουν σε ένα τύπο που ορίζουμε εμείς στο WSDL αρχείο. Για παράδειγμα στο συγκεκριμένο WSDL αρχείο ορίζουμε το namespace xsd1=http://sorch.netmode.ntua.gr/ws/ έτσι ώστε τα καινούργια types που εισάγουμε να αναφέρονται σε αυτό το namespace.

- **types**

Στο στοιχείο αυτό περιλαμβάνονται οι σύνθετοι τύποι δεδομένων που ανταλλάσσονται μεταξύ του πελάτη και του εξυπηρετητή. Αν έχουμε μόνο απλούς τύπους όπως string ή integer τότε μπορούμε να παραλείψουμε αυτό το στοιχείο. Μας δίνεται η δυνατότητα να ορίσουμε μέσα στο WSDL αρχείο σύνθετους τύπους δεδομένων. Για παράδειγμα στο συγκεκριμένο αρχείο έχουμε ορίσει τον τύπο RSpec, ο οποίος είναι σύνθετου τύπου (complex type). Ο τύπος αυτός περιλαμβάνει σαν στοιχεία «παιδιά» τα name, owner, timestamp, duration και rspec, τα οποία είναι απλού τύπου είτε string είτε int, για την περιγραφή του slice. Το name είναι το όνομα του slice, ο owner είναι ο χρήστης, το timestamp δηλώνει τη χρονική στιγμή που ξεκινάει το activation του slice, το duration υποδηλώνει τη χρονική διάρκεια του activation σε ώρες και το rspec είναι η περιγραφή σε RSpec μορφή, όπως θα αναλύσουμε στο επόμενο κεφάλαιο, των διάφορων κόμβων του slice. Όπως αναφέραμε και προηγουμένως ο καινούργιος αυτός τύπος που ορίζεται αναφέρεται στο δικό μας namespace `xsd1=http://sorch.netmode.ntua.gr/ws/` και μπορούμε να τον χρησιμοποιήσουμε σαν παράμετρο στα μηνύματα που ανταλλάσσονται, όπως θα δούμε παρακάτω.

- **message**

Στο παράδειγμα έχουμε τέσσερα στοιχεία message, δύο για κάθε λειτουργία. Όπως βλέπουμε το ένα από τα δύο στοιχεία κάθε λειτουργίας είναι μήνυμα αίτησης για την συγκεκριμένη λειτουργία και το δεύτερο είναι μήνυμα απάντησης ως αποτέλεσμα της λειτουργίας. Από το χαρακτηριστικό type του στοιχείου part καταλαβαίνουμε τους τύπους των δεδομένων που ανταλλάσσονται. Στο συγκεκριμένο παράδειγμα τα μηνύματα αίτησης έχουν παράμετρο τύπου RSpec που ορίσαμε εμείς στο WSDL αρχείο και τα μηνύματα απόκρισης έχουν παράμετρο τύπου string που ουσιαστικά ενημερώνει το χρήστη για τις διεργασίες που πραγματοποιήθηκαν.

Αν μια λειτουργία περιμένει πολλαπλά ορίσματα ή επιστρέφει πολλαπλές τιμές, τότε μπορούμε να ορίσουμε για κάθε ένα από αυτά ένα στοιχείο part.

- **portType**

Το στοιχείο portType του παραδείγματος μας περιλαμβάνει δύο λειτουργίες τη mkslice και την modifyslice. Η καθεμία από τις δύο λειτουργίες έχει εάν input message και ένα output message τα οποία περιγράψαμε παραπάνω. Η αναφορά σε αυτά γίνεται με χρήση κατάλληλου namespace και συγκεκριμένα του `tns=http://sorch.netmode.ntua.gr/ws/` που έχει οριστεί στο στοιχείο definitions. Με αυτό τον τρόπο το σύστημα αναγνωρίζει τι τύπου θα είναι τα διάφορα μηνύματα και τι παραμέτρους θα εμπεριέχουν.

- **binding**

Το στοιχείο binding παρέχει ειδικές λεπτομέρειες για το πώς μια λειτουργία που καθορίζεται από το στοιχείο portType θα μεταφερθεί πάνω στο καλώδιο. Οι

συνδέσεις μπορούν να γίνουν μέσω διαφόρων πρωτοκόλλων, συμπεριλαμβανομένων των HTTP GET, HTTP POST ή SOAP. Στην πραγματικότητα μπορείς να ορίσεις πολλές συνδέσεις (bindings) για κάθε στοιχείο portType. Στο παράδειγμα μας για την μεταφορά των λειτουργιών χρησιμοποιείται RPC over HTTP.

- **service**

Το στοιχείο service καθορίζει τη διεύθυνση για την κλήση της συγκεκριμένης υπηρεσίας και περιέχει το URL της web service. Στο παράδειγμά μας το URL της υπηρεσίας διαδικτύου είναι 'http://sorch.netmode.ntua.gr/ws/mksliceservice.php'.

2.7.4 Περιγραφή WSO2 WSF extension for php web services

Ένα σημαντικό εργαλείο που χρησιμοποιήσαμε για την υλοποίηση κάποιων υπηρεσιών στην παρούσα διπλωματική είναι το wsf extension για php [29], το οποίο μας διευκόλυνε σημαντικά στην δημιουργία των υπηρεσιών διαδικτύου βασισμένων σε php. Το framework αυτό κάνει εύκολο το σχεδιασμό των web services αφού περιλαμβάνει πολλές έτοιμες κλάσεις για τη δημιουργία των μηνυμάτων που ανταλλάσσονται, της υπηρεσίας, του πελάτη καθώς και για θέματα ασφαλείας. Τέτοιες κλάσεις είναι οι WSClnt, WSClntProxy, WSService, WSMMessage, WSHeader, WSFault, WSSecurityToken, WSPolicy, οι οποίες μπορούν να χρησιμοποιηθούν στο php πρόγραμμα εφόσον έχει εγκατασταθεί το extension στο μηχάνημα μας, όπως αναφέρεται στο [30].

Ένα άλλο σημαντικό πλεονέκτημα που μας προσφέρει αυτό το framework είναι ότι μπορούμε να δουλέψουμε είτε σε WSDL mode είτε σε NO-WSDL mode ([30]). Στην πρώτη περίπτωση χρησιμοποιούμε ένα wsdl αρχείο το οποίο όπως αναφέραμε θα έχει την περιγραφή της υπηρεσίας, των λειτουργιών της και των διάφορων τύπων δεδομένων που ανταλλάσσονται. Χρησιμοποιώντας αυτό το αρχείο θα γίνεται η επικοινωνία μεταξύ του πελάτη και του εξυπηρετητή, δηλαδή ο client θα καλεί το operation της υπηρεσίας που επιθυμεί. Στη δεύτερη περίπτωση δεν χρησιμοποιούμε κάποιο wsdl αρχείο για την περιγραφή των υπηρεσιών αλλά ο client διαμορφώνει το μήνυμα (WSMessage) κατά τέτοιο τρόπο ώστε να αναφέρεται σε μια λειτουργία μιας συγκεκριμένης υπηρεσίας, η οποία καθορίζεται με το κατάλληλο URL. Μέσα στο μήνυμα θα πρέπει να υπάρχουν οι κατάλληλες παράμετροι για τη συγκεκριμένη λειτουργία. Με αυτή τη δυνατότητα που παρέχει το wsf extension κάνει πιο απλή στο χρήστη τη σχεδίαση των υπηρεσιών, αφού δεν χρειάζεται να δημιουργήσει wsdl αρχείο. Το wsdl αρχείο της υπηρεσίας μπορεί να γίνει generated στη συνέχεια πληκτρολογώντας το URL της υπηρεσίας μαζί με ?wsdl. Αξίζει να σημειωθεί ότι η λειτουργία χωρίς WSDL αρχείο είναι δύσκολο να χρησιμοποιηθεί όταν θέλουμε να ανταλλάξουμε τύπους δεδομένων σύνθετου τύπου ή πίνακες. Για αυτό το λόγο στην παρούσα διπλωματική χρησιμοποιήθηκε μόνο από υπηρεσίες που χρησιμοποιούσαν απλούς τύπους δεδομένων, όπως string ή integer.

Παραθέτουμε ένα παράδειγμα ενός μηνύματος WSMMessage που στέλνει ο client στο server για να δούμε τι πρέπει να περιλαμβάνει:

Κεφάλαιο 2: Βασικά Πρωτόκολλα και Τεχνολογίες

```
$reqPayloadString1 = <<<XML
<ns1:delslice xmlns:ns1="http://sorch.netmode.ntua.gr/ws/delsliceservice.php">
<name>$name</name>
</ns1:delslice>
XML;

$reqMessage1 = new WSMMessage($reqPayloadString1);
```

Πίνακας 2.7: Παράδειγμα WSMMessage

Στο παραπάνω παράδειγμα παρατηρούμε ότι ο client θέλει να καλέσει τη λειτουργία `delslice` που ανήκει στην υπηρεσία με URL `http://sorch.netmode.ntua.gr/ws/delsliceservice.php` με παράμετρο το όνομα του `slice`, προκειμένου να διαγράψει το `slice` με αυτό το όνομα. Βλέπουμε λοιπόν ότι ο χρήστης διαμορφώνει κατάλληλα το `WSMessage` για να καλέσει το `operation` της υπηρεσίας που θέλει με τις αντίστοιχες παραμέτρους στο σώμα του `message`.

Κεφάλαιο 3

Μοντέλα Περιγραφής Διαμοιραζόμενων Πόρων

3.1 Resource Specification (RSpec)

Η περιγραφή των πόρων (Resource Description) παίζει σημαντικό ρόλο στην επικοινωνία και αλληλεπίδραση τόσο μεταξύ των διάφορων πειραματικών πλατφορμών (testbeds) όσο και μεταξύ των χρηστών και των μονάδων διαχείρισης (component manager) των πόρων των διάφορων πειραματικών πλατφορμών. Ως εκ τούτου έχουν αναπτυχθεί αρκετές δομές δεδομένων και μοντέλα πληροφορίας για την κατάλληλη περιγραφή των διαμοιραζόμενων πόρων. Σε αυτές τις δομές δεδομένων συγκαταλέγεται και το Resource Specification (RSpec) των πόρων.

3.1.1 Περιγραφή του RSpec

Το Resource Specification (RSpec) είναι ένα XML κείμενο το οποίο χρησιμοποιείται από τις διάφορες πειραματικές πλατφόρμες για την ανταλλαγή πληροφοριών σχετικά με τους πόρους τους και πρέπει να είναι ανεξάρτητο της εκάστοτε υλοποίησης τόσο σε υλικό όσο και σε λογισμικό [1]. Με τη χρήση του RSpec μπορούν να επιστρέφονται πληροφορίες σε κατάλληλη δομή στο χρήστη αναφορικά με το ποιό πόροι είναι διαθέσιμοι και στη συνέχεια ο χρήστης διαμορφώνοντας κατάλληλο RSpec αρχείο να διευκρινίζει ποιούς πόρους και με τι χαρακτηριστικά επιθυμεί να δεσμεύσει για τα πειράματα του και να δημιουργήσει το slice του. Το RSpec θα πρέπει να περιέχει κατάλληλα στοιχεία όπως για παράδειγμα το όνομα του κόμβου, της διεπαφής, τα οποία όπως θα δούμε καθορίζονται από τα αντίστοιχα URNs, τον τύπο του μηχανήματος (hardware-type) καθώς και άλλες πληροφορίες που θα εξετάσουμε ακολούθως.

Το RSpec θα πρέπει να χρησιμοποιηθεί για να παρέχει τριών ειδών πληροφορίες. Αρχικά θα πρέπει να παρέχει πληροφορίες σχετικά με τους πόρους της πειραματικής πλατφόρμας που είναι διαθέσιμοι για να τους δεσμεύσει ο χρήστης και να δημιουργήσει το slice του (RSpec Advertisement [2]). Στη συνέχεια θα πρέπει να παρέχει μια περιγραφή για τους πόρους τους οποίους ζητάει ο χρήστης για να πραγματοποιήσει τα πειράματα του (RSpec Request [2]). Τέλος θα πρέπει να περιγράφει τους πόρους που έχουν δεσμευθεί από κάποιον χρήστη για να πραγματοποιήσει τα πειράματα του και τα οποία ανήκουν σε κάποιο slice (RSpec Manifest [2]). Τα τρία αυτά είδη RSpec θα τα αναλύσουμε περισσότερο σε επόμενο υποκεφάλαιο.

Επιπλέον επειδή το RSpec αποτελεί ένα καλώς δομημένο XML κείμενο για την περιγραφή των πόρων θα πρέπει να συμμορφώνεται με κάποια κατάλληλα XML σχήματα, τα οποία βασίζονται στις δομές δεδομένων που έχουν αναπτυχθεί για την περιγραφή των χαρακτηριστικών των πόρων. Ως εκ τούτου ένα εξίσου σημαντικό κομμάτι που θα εξετάσουμε σε αυτή τη διπλωματική είναι τα RSpec αρχεία ανάλογα με το είδος στο οποίο ανήκουν να συμμορφώνονται με το αντίστοιχο XML σχήμα.

Έτσι οι πληροφορίες που ανταλλάσσονται θα έχουν μια καλώς δομημένη μορφή και θα μπορούν να επεξεργαστούν χρησιμοποιώντας κατάλληλους XML αναλυτές σύνταξης (parsers)..

3.1.2 Πλεονεκτήματα του RSpec

Το RSpec χρησιμοποιείται ευρέως από τις διάφορες πειραματικές πλατφόρμες για την περιγραφή των resources και των διαφόρων χαρακτηριστικών τους. Αυτό οφείλεται στο γεγονός ότι το RSpec και η δομή του έχει πολλά πλεονεκτήματα. Μερικά από αυτά είναι:

- Συμβάλλει στην προσπάθεια ενοποίησης των διάφορων πειραματικών πλατφορμών (federated network testbeds) έτσι ώστε οι ερευνητές να μπορούν να χρησιμοποιήσουν πόρους που ανήκουν σε διαφορετικές πειραματικές πλατφόρμες για τα πειράματά τους. Αν διαμορφωθεί ένα κοινό σχήμα για το RSpec που θα χρησιμοποιείται από ετερογενείς πειραματικές πλατφόρμες τότε ο ερευνητής μπορεί να εκμεταλλευτεί τους πόρους που ανήκουν σε όλες αυτές τις πλατφόρμες και να ζητήσει τους πόρους με τα αντίστοιχα χαρακτηριστικά που επιθυμεί, διαμορφώνοντας το RSpec κατάλληλα.

- Παρέχει ένα καθορισμένο τρόπο περιγραφής των χαρακτηριστικών των πόρων σε μορφή XML κειμένου, ο οποίος είναι ανεξάρτητος από το λογισμικό που χρησιμοποιεί το εκάστοτε μηχάνημα. Αυτό είναι πολύ σημαντικό αφού γνωρίζοντας την δομή του RSpec μπορούμε να εξαγάγουμε τις πληροφορίες που χρειαζόμαστε για τους πόρους είτε στην πλευρά του χρήστη είτε του εξυπηρετητή χρησιμοποιώντας απλά κάποιον XML Parser.

- Η δομή του RSpec v.2 [2] έχει σχεδιαστεί κατά τέτοιο τρόπο ώστε να είναι επεκτάσιμη για εισαγωγή καινούργιων χαρακτηριστικών. Όπως θα δούμε και σε παρακάτω υποκεφάλαιο υπάρχει ένα σχήμα κορμού για το RSpec v.2 (Core RSpec) ανάλογα με το είδος στο οποίο ανήκει, το οποίο περιέχει τα απαραίτητα στοιχεία, και επιπλέον έχουμε τη δυνατότητα να προσθέσουμε νέα είδη πόρων ή καινούργια χαρακτηριστικά στους ήδη υπάρχοντες δημιουργώντας ένα σχήμα επέκτασης (extension schema). Αξίζει να επισημάνουμε ότι το σχήμα επέκτασης θα πρέπει να έχει διαφορετικό πεδίο ονομάτων (namespace) από το Core RSpec. Το γεγονός ότι το RSpec v.2 είναι επεκτάσιμο μας δίνει τη δυνατότητα να μπορούμε να πραγματοποιήσουμε μεγάλη πληθώρα πειραμάτων με πόρους που έχουν χαρακτηριστικά τα οποία ενδέχεται να μην ανήκουν στο σχήμα κορμού. Αυτό το εκμεταλλευτήκαμε και στην παρούσα διπλωματική για να περιγράψουμε τα χαρακτηριστικά των ασύρματων κόμβων. Συγκεκριμένα προσθέσαμε τα απαραίτητα χαρακτηριστικά που πρέπει να έχουν οι ασύρματες διεπαφές των κόμβων, όπως είναι ρυθμός μετάδοσης, κανάλι μετάδοσης και λειτουργία των διεπαφών, τα οποία θα αναλύσουμε σε επόμενο υποκεφάλαιο.

- Το RSpec επειδή έχει XML δομή μπορεί να ελεγχθεί ως προς την εγκυρότητα του (validation) με βάση κάποια σχήματα. Τα σχήματα αυτά περιέχουν τα απαραίτητα χαρακτηριστικά που πρέπει να έχει το RSpec με βάση τον τύπο στον οποίο ανήκει. Επιπλέον επειδή το RSpec παρέχει τη δυνατότητα να δημιουργήσουμε επεκτάσεις, οι οποίες θα έχουν διαφορετικό πεδίο ονομάτων από το core RSpec, δημιουργείται η ανάγκη για έλεγχο εγκυρότητας με δυο ή περισσότερα σχήματα (Core RSpec και extension schemas). Αυτό επιτυγχάνεται με χρήση της γλώσσας NVDL που αναλύσαμε παραπάνω και διάφορων εργαλείων που έχουν αναπτυχθεί σε γλώσσες προγραμματισμού για την εφαρμογή αυτής της γλώσσας σε XML κείμενα, έτσι ώστε να ελέγξουμε την εγκυρότητα τους. Ως εκ τούτου έχουμε τη δυνατότητα να ελέγξουμε την εγκυρότητα των RSpec αρχείων που ανταλλάσσονται με βάση τόσο Core RSpec όσο και τις διάφορες επεκτάσεις που χρησιμοποιούνται από την πειραματική πλατφόρμα. Το γεγονός αυτό ήταν πολύ σημαντικό για την υλοποίηση της διπλωματικής αφού τα RSpec έπρεπε να συμμορφώνονται τόσο με ένα βασικό σχήμα όσο και με την επέκταση που δημιουργήσαμε για τα χαρακτηριστικά των ασύρματων διεπαφών ώστε να γίνουν αποδεκτά και να επιτραπεί η περαιτέρω επεξεργασία τους.

3.1.3 Βασικό σχήμα του RSpec v.2 (Core RSpec) και κατηγορίες RSpec v.2

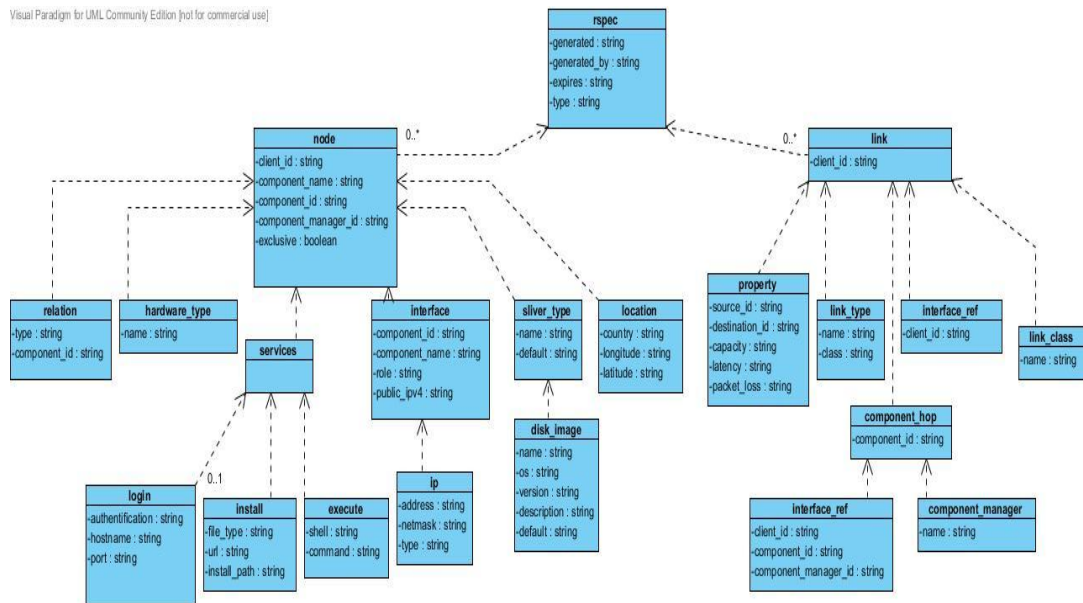
Το RSpec βασίζεται σε κάποια σχήματα κορμού (Core RSpec) έτσι ώστε να περιέχει τα βασικά χαρακτηριστικά για την περιγραφή των πόρων των πειραματικών πλατφορμών. Τα σχήματα κορμού είναι διαφορετικά ανάλογα με την κατηγορία στην οποία ανήκει το RSpec. Οι κατηγορίες των RSpec είναι τρεις όπως αναφέραμε και προηγουμένως το RSpec request [2], το RSpec advertisement [2] και το RSpec manifest [2] και τα αντίστοιχα σχήματα περιέχουν τα στοιχεία που είναι απαραίτητα για κάθε ένα από αυτά τα είδη RSpec.

Αξίζει να επισημάνουμε ότι τα Core RSpec v.2 schemas που χρησιμοποιήθηκαν σε αυτή τη διπλωματική είναι τα σχήματα που χρησιμοποιεί το ProtoGENI [2] για τις τρεις αυτές κατηγορίες του RSpec. Επιπλέον για την περιγραφή των πόρων της πειραματικής πλατφόρμας κάναμε χρήση των URNs που χρησιμοποιεί ως αναγνωριστικά των πόρων και το ProtoGENI.

3.1.3.1 RSpec v.2 Request

Το RSpec v.2 Request, όπως αναφέρεται στο [2], είναι ένα είδος RSpec που διαμορφώνει ο χρήστης για να δηλώσει ποιά resources και με τι χαρακτηριστικά θέλει να δεσμεύσει για να πραγματοποιήσει τα πειράματά του. Το RSpec αυτό θα πρέπει να συμμορφώνεται με ένα σχήμα κορμού που καθορίστηκε από το ProtoGENI έτσι ώστε να περιέχει τα απαραίτητα χαρακτηριστικά για την περιγραφή των πόρων που ο χρήστης επιθυμεί να δεσμεύσει. Παραθέτουμε το UML διάγραμμα αυτού του σχήματος, όπως περιγράφεται στο [2]:

Κεφάλαιο 3: Μοντέλα Περιγραφής Διαμοιραζόμενων Πόρων



Σχήμα 3.1: UML διάγραμμα του RSpec v.2 Request [2]

Απο το παραπάνω διάγραμμα παρατηρούμε ότι το RSpec τύπου Request θα πρέπει να περιέχει κάποιους κόμβους (nodes) ή και κάποιες ζεύξεις (links) της πειραματικής πλατφόρμας, τα οποία καθορίζονται με βάση το `client_id` που είναι μοναδικό για κάθε κόμβο ή ζεύξη. Επιπλέον όπως βλέπουμε στην περιγραφή του κάθε κόμβου μπορεί να εμπεριέχεται ποίες διεπαφές (interfaces) του κόμβου θέλει να χρησιμοποιήσει ο χρήστης και με τι χαρακτηριστικά ή για παράδειγμα ποια services του κόμβου θέλει να χρησιμοποιήσει ο ερευνητής καθώς και άλλα χαρακτηριστικά. Τέτοια χαρακτηριστικά είναι η τοποθεσία του κόμβου (location), ο τύπος του μηχανήματος (hardware_type) και το image που τρέχει στον κόμβο (disk_image), το οποίο περιλαμβάνει το τρέχον λειτουργικό σύστημα. Ομοίως βλέπουμε ότι μέσα στην περιγραφή του link μπορεί να περιέχονται άλλα χαρακτηριστικά. Έχοντας αυτή την δεδομένη δομή με την οποία πρέπει να συμμορφώνεται το RSpec request ο ερευνητής έχει τη δυνατότητα να περιγράψει τα βασικά χαρακτηριστικά των resources που θέλει να δεσμεύσει και να στείλει το RSpec στον εξυπηρετητή για περαιτέρω επεξεργασία των πληροφοριών.

Ας εξετάσουμε ένα απλό παράδειγμα RSpec v.2 Request του ProtoGENI, το οποίο πήραμε από το [2], που βασίζεται σε αυτό το σχήμα και περιγράφει ένα συγκεκριμένο κόμβο που θέλει ο ερευνητής:

```
<?xml version="1.0" encoding="UTF-8"?>
<rspec xmlns="http://www.protogeni.net/resources/rspec/2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.protogeni.net/resources/rspec/2
    http://www.protogeni.net/resources/rspec/2/request.xsd"
  type="request">
  <node component_id="urn:publicid:IDN+emulab.net+node+pc175"
```



```
component_manager_id="urn:publicid:IDN+emulab.net+authority+cm"  
client_id="pc175"  
exclusive="true">  
<sliver_type name="raw-pc" />  
</node>  
</rspec>
```

Πίνακας 3.1: Παράδειγμα RSpec v.2 Request

Το παραπάνω παράδειγμα είναι ένα RSpec v.2 Request για ένα συγκεκριμένο κόμβο της πειραματικής πλατφόρμας και συμμορφώνεται όπως βλέπουμε από το schemaLocation του rspec με το Core RSpec schema για το request, το οποίο παραθέσαμε παραπάνω. Ο ερευνητής επιθυμεί ένα συγκεκριμένο κόμβο το pc175. Όπως βλέπουμε ο κόμβος αυτός καθορίζεται με βάση το URN του urn:publicid:IDN+emulab.net+node+pc175 στο πεδίο (attribute) component_id. Επιπλέον από το attribute exclusive="true" καταλαβαίνουμε ότι ο χρήστης θέλει αποκλειστική χρήση του συγκεκριμένου κόμβου (reservable node).

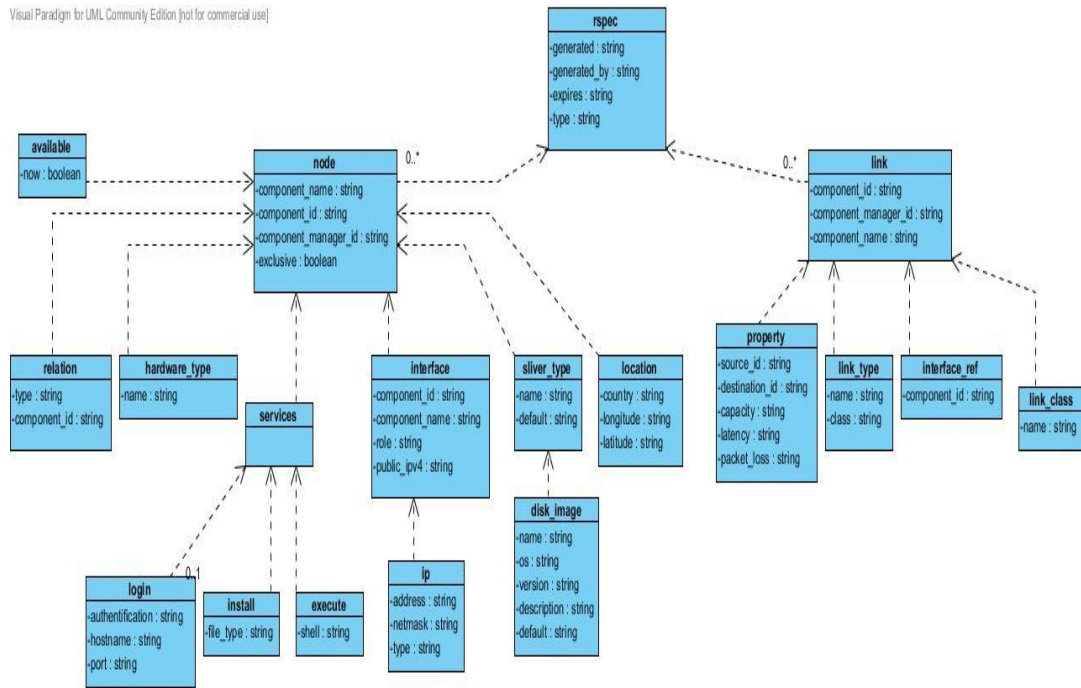
3.1.3.2 RSpec v.2 Advertisement

Το RSpec v.2 Advertisement, όπως αναφέρεται στο [2], είναι ένα είδος RSpec που δημιουργείται από το εργαλείο διαχείρισης της πειραματικής πλατφόρμας για να ενημερώσει τους ερευνητές σχετικά με το ποιό πόροι είναι διαθέσιμοι εκείνη τη χρονική στιγμή που θέλουν να πραγματοποιήσουν τα πειράματά τους. Αυτό το RSpec πρέπει να περιέχει τα κατάλληλα χαρακτηριστικά για να βοηθήσουν τον ερευνητή να επιλέξει τους πόρους που επιθυμεί. Έτσι με βάση το RSpec Advertisement ο χρήστης ενημερώνεται για το τι είναι διαθέσιμο στην πειραματική πλατφόρμα και στη συνέχεια μπορεί να διαμορφώσει το RSpec Request και να το στείλει στον εξυπηρετητή για να δεσμεύσει τους αντίστοιχους πόρους.

Το RSpec v.2 Advertisement θα πρέπει να συμμορφώνεται με ένα σχήμα κορμού που καθορίστηκε από το ProtoGENI έτσι ώστε να περιέχει τα απαραίτητα χαρακτηριστικά για να μπορεί ο χρήστης να ενημερωθεί κατάλληλα για τους πόρους που είναι διαθέσιμοι. Παραθέτουμε το UML διάγραμμα αυτού του σχήματος, όπως περιγράφεται στο [2]:

Κεφάλαιο 3: Μοντέλα Περιγραφής Διαμοιραζόμενων Πόρων

Visual Paradigm for UML Community Edition (not for commercial use)



Σχήμα 3.2: UML διάγραμμα του RSpec v.2 Advertisement [2]

Από το παραπάνω διάγραμμα παρατηρούμε ότι το RSpec τύπου Advertisement θα πρέπει να περιέχει κάποιους κόμβους (nodes) ή και κάποιες ζεύξεις (links) της πειραματικής πλατφόρμας, τα οποία καθορίζονται με βάση το υποχρεωτικό πεδίο component_id που περιέχει το URN αυτών των στοιχείων είναι μοναδικό για κάθε κόμβο ή ζεύξη. Επιπλέον όπως βλέπουμε στην περιγραφή του κάθε κόμβου μπορεί να εμπεριέχεται ποιές διεπαφές (interfaces) του κόμβου και με ποια χαρακτηριστικά είναι διαθέσιμες ή για παράδειγμα ποιές υπηρεσίες του κόμβου μπορεί να χρησιμοποιήσει ο ερευνητής. Τέτοια χαρακτηριστικά είναι η τοποθεσία του κόμβου (location), ο τύπος του μηχανήματος (hardware_type) και το image που τρέχει στον κόμβο (disk_image), το οποίο περιλαμβάνει το τρέχον λειτουργικό σύστημα. Ομοίως βλέπουμε ότι μέσα στην περιγραφή της ζεύξης (link) μπορεί να περιέχονται άλλα χαρακτηριστικά. Έχοντας αυτή την δεδομένη δομή με την οποία πρέπει να συμμορφώνεται το RSpec advertisement μπορούμε να ενημερωθούμε για τα στοιχεία της πειραματικής πλατφόρμας που είναι διαθέσιμα για ένα δεδομένο χρονικό διάστημα.

Ας εξετάσουμε ένα απλό παράδειγμα RSpec v.2 Advertisement του ProtoGENI, το οποίο πήραμε από το [2], που βασίζεται σε αυτό το σχήμα και περιγράφει τα στοιχεία ενός κόμβου που είναι διαθέσιμος:

```
<?xml version="1.0" encoding="UTF-8"?>
<rspec xmlns="http://www.protogeni.net/resources/rspec/2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.protogeni.net/resources/rspec/2
http://www.protogeni.net/resources/rspec/2/ad.xsd"
  type="advertisement"
  generated="2009-07-21T19:19:06Z"
  valid_until="2009-07-21T19:19:06Z" >
```

Κεφάλαιο 3: Μοντέλα Περιγραφής Διαμοιραζόμενων Πόρων

```
<node component_manager_id="urn:publicid:IDN+emulab.geni.emulab.net+authority+cm"
  component_name="pc160"
  component_id="urn:publicid:IDN+emulab.geni.emulab.net+node+pc160" >

  <available>true</available>
  <exclusive>true</exclusive>
  <interface component_id="urn:publicid:IDN+emulab.geni.emulab.net+interface+pc160:eth0"/>
  <interface component_id="urn:publicid:IDN+emulab.geni.emulab.net+interface+pc160:eth1"/>

</node>
</rspec>
```

Πίνακας 3.2: Παράδειγμα RSpec v.2 Advertisement

Το παραπάνω παράδειγμα είναι RSpec v.2 Advertisement που περιγράφει ένα διαθέσιμο κόμβο του ProtoGENI και συμμορφώνεται όπως βλέπουμε από το schemaLocation του rspec με το Core RSpec schema για το advertisement, το οποίο παραθέσαμε παραπάνω. Παρατηρούμε ότι ο κόμβος περιγράφεται από το URN του urn:publicid:IDN+emulab.geni.emulab.net+node+pc160 στο πεδίο component_id. Επιπλέον από τα στοιχεία available και exclusive βλέπουμε ότι ο κόμβος είναι διαθέσιμος στους ερευνητές και μπορεί να χρησιμοποιηθεί αποκλειστικά από έναν ερευνητή για τα πειράματα του. Ακόμα μας δίνεται η πληροφορία ότι ο κόμβος έχει δύο διαθέσιμες διεπαφές (interface), οι οποίες περιγράφονται από τα αντίστοιχα URNs τους (urn:publicid:IDN+emulab.geni.emulab.net+interface+pc160:eth0 και urn:publicid:IDN+emulab.geni.emulab.net+interface+pc160:eth1). Ως εκ τούτου ο ερευνητής μπορεί να επιλέξει και κάποιες από τις διαθέσιμες διεπαφές του κόμβου για να χρησιμοποιήσει στα πειράματα του.

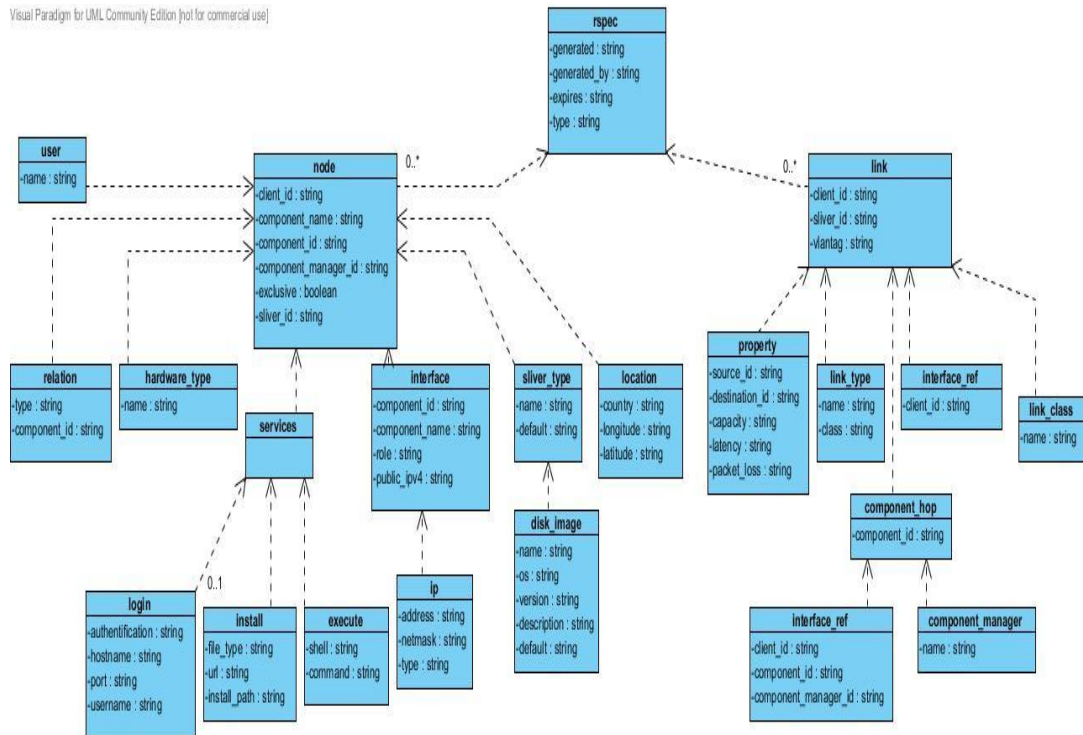
3.1.3.3 RSpec v.2 Manifest

Το RSpec v.2 Manifest, όπως αναφέρεται στο [2], είναι ένα είδος RSpec που δημιουργείται από τη διαχειριστική οντότητα για να ενημερώσει σχετικά με το ποιοι πόροι είναι δεσμευμένοι από κάποιον ερευνητή.

Το RSpec v.2 Manifest, όπως και τα άλλα είδη RSpec, θα πρέπει να συμμορφώνεται με ένα σχήμα κορμού που καθορίστηκε από το ProtoGENI έτσι ώστε να περιέχει τα απαραίτητα χαρακτηριστικά για να περιγραφούν οι πόροι που έχουν δεσμευθεί. Παραθέτουμε το UML διάγραμμα αυτού του σχήματος, όπως περιγράφεται στο [2]:

Κεφάλαιο 3: Μοντέλα Περιγραφής Διαμοιραζόμενων Πόρων

Visual Paradigm for UML Community Edition (not for commercial use)



Σχήμα 3.3: UML διάγραμμα του RSpec v.2 Manifest [2]

Από το παραπάνω διάγραμμα παρατηρούμε ότι το RSpec τύπου Manifest θα πρέπει να περιέχει κάποιους κόμβους (nodes) ή και κάποιες ζεύξεις (links) της πειραματικής πλατφόρμας, τα οποία ανήκουν σε δεδομένο slice ενός ερευνητή και καθορίζονται με βάση το υποχρεωτικό πεδίο component_id που περιέχει το URN αυτών των στοιχείων και είναι μοναδικό για κάθε κόμβο ή ζεύξη. Επιπλέον μπορεί να εμπεριέχεται ποιές διεπαφές (interfaces) του κόμβου χρησιμοποιεί ο χρήστης και με τι χαρακτηριστικά ή για παράδειγμα ποιές υπηρεσίες του κόμβου χρησιμοποιεί ο ερευνητής καθώς και άλλα χαρακτηριστικά. Τέτοια χαρακτηριστικά είναι η τοποθεσία του κόμβου (location), ο τύπος του μηχανήματος (hardware_type) και το image που τρέχει στον κόμβο (disk_image), το οποίο περιλαμβάνει το τρέχον λειτουργικό σύστημα. Ομοίως βλέπουμε ότι μέσα στην περιγραφή της ζεύξης μπορεί να περιέχονται άλλα χαρακτηριστικά. Έχοντας αυτή την δεδομένη δομή με την οποία πρέπει να συμμορφώνεται το RSpec v.2 Manifest ο ερευνητής έχει τη δυνατότητα να περιγράψει τα βασικά χαρακτηριστικά των resources που έχει εκχωρήσει σε ένα δεδομένο slice του.

Ας εξετάσουμε ένα απλό παράδειγμα RSpec v.2 Manifest του ProtoGENI, το οποίο πήραμε από το [2], που βασίζεται σε αυτό το σχήμα και περιγράφει τα στοιχεία που έχουν εκχωρηθεί σε ένα δεδομένο slice του χρήστη:

```
<?xml version="1.0" encoding="UTF-8"?>
<rspec xmlns="http://www.protogeni.net/resources/rspec/2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.protogeni.net/resources/rspec/2
  http://www.protogeni.net/resources/rspec/2/manifest.xsd"
  type="manifest"
  generated="2009-07-21T19:19:06Z"
```

```

valid_until="2009-07-21T19:19:06Z" >

<node component_manager_id="urn:publicid:IDN+emulab.geni.emulab.net+authority+cm"
  component_name="pc160"
  component_id="urn:publicid:IDN+emulab.geni.emulab.net+node+pc160"
  sliver_id="de980246-773e-102b-8eb4-001143e453fe"
  exclusive="true">

  <sliver_type name="raw-pc" />
  <interface client_id="pc160:eth0"
    component_id="urn:publicid:IDN+emulab.geni.emulab.net+interface+pc160:eth0/>
</node>

<node component_manager_id="urn:publicid:IDN+emulab.geni.emulab.net+authority+cm"
  component_name="pc162"
  component_id="urn:publicid:IDN+emulab.geni.emulab.net+node+pc162"
  sliver_id="de989c2d-773e-102b-8eb4-001143e453fe"
  exclusive="true">

  <sliver_type name="raw-pc" />
  <interface client_id="pc162:eth0"
    component_id="urn:publicid:IDN+emulab.geni.emulab.net+interface+pc162:eth0/>
</node>
</rspec>

```

Πίνακας 3.3: Παράδειγμα RSpec v.2 Manifest

Το παραπάνω παράδειγμα είναι RSpec v.2 Manifest που περιγράφει δύο κόμβους του ProtoGENI που ανήκουν σε κάποιο slice του ερευνητή και συμμορφώνεται όπως βλέπουμε στο schemaLocation του rspec με Core RSpec schema για το manifest. Παρατηρούμε ότι ο κάθε κόμβος περιγράφεται από το URN του, όπως για παράδειγμα το urn:publicid:IDN+emulab.geni.emulab.net+node+pc160 στο πεδίο component_id. Ακόμα μας δίνεται πληροφορία για τις ασύρματες διεπαφές (interface) που έχουμε δεσμεύσει από κάθε κόμβο. Όπως βλέπουμε αυτές περιγράφονται από τα αντίστοιχα URNs τους που έχουν κατάλληλη δομή ώστε να περιγράψουν τις ασύρματες διεπαφές, αφού έχουν interface ως τύπο resource. Ένα παράδειγμα τέτοιου τύπου URN είναι urn:publicid:IDN+emulab.geni.emulab.net+interface+pc160:eth0, το οποίο περιγράφει τη διεπαφή eth0 του κόμβου pc160 του ProtoGENI και μας ενημερώνει ότι αυτό το interface ανήκει στο συγκεκριμένο slice του ερευνητή. Συνεπώς με τη χρήση του RSpec v.2 Manifest ενημερώνονται οι χρήστες για τους πόρους της πειραματικής πλατφόρμας που έχουν εκχωρηθεί σε κάποιο slice.

3.2 Επέκταση RSpec για περιγραφή ασύρματων κόμβων

Στην παρούσα διπλωματική εκμεταλλευτήκαμε τη δυνατότητα που μας παρέχει το Core RSpec v.2 schema για δημιουργία μιας επέκτασης για την κατάλληλη περιγραφή χαρακτηριστικών και πόρων που δεν περιλαμβάνονται στο Core Schema του RSpec v.2. Η επέκταση αυτή θα πρέπει να έχει διαφορετικό πεδίο ονομάτων από το Core Schema.

Η εργαστηριακή πλατφόρμα του Network Management & Optimal Design Laboratory (NTUA), όπως περιγράψαμε και στο πρώτο κεφάλαιο, περιλαμβάνει 10 ασύρματους κόμβους, ο καθένας από τους οποίους διαθέτει δύο ασύρματες διεπαφές (interfaces) που μπορεί να χρησιμοποιήσει ο χρήστης για τα πειράματά του. Ετσι είχαμε την ανάγκη να βρούμε ένα τρόπο περιγραφής ασύρματων πόρων και των διάφορων χαρακτηριστικών τους έτσι ώστε να χρησιμοποιήσουμε το RSpec v.2 στην πειραματική μας πλατφόρμα, μιας και δεν υποστήριζε ασύρματους πόρους. Στην επέκταση που δημιουργήσαμε εισαγάγαμε τα απαραίτητα χαρακτηριστικά για την περιγραφή ασύρματων κόμβων.

3.2.1 Χαρακτηριστικά ασύρματων κόμβων

Τα βασικά χαρακτηριστικά για να περιγράψουμε ασύρματους κόμβους είναι εκείνα που περιγράφουν τις ασύρματες διεπαφές (wireless interfaces) του κόμβου, μέσω των οποίων επικοινωνεί και ανταλλάσσει πληροφορίες με το υπόλοιπο δίκτυο. Οι ασύρματες διεπαφές μπορούν να περιγραφούν κατάλληλα από τις παρακάτω οντότητες, τις οποίες χρησιμοποιήσαμε στο μοντέλο δεδομένων (data model) που δημιουργήσαμε και το οποίο χρησιμοποιήθηκε ως επέκταση του RSpec v.2:

- **Ρυθμός Μετάδοσης (Rate)**

Ο ρυθμός μετάδοσης μας δείχνει τον αριθμό των bits πληροφορίας ανα δευτερόλεπτο (bits/sec ή bps), τα οποία είτε μεταδίδονται είτε λαμβάνονται από την ασύρματη διεπαφή. Αποτελεί βασικό μέγεθος για την περιγραφή της επικοινωνίας μεταξύ ασύρματων διεπαφών. Ο ρυθμός μετάδοσης είναι ανάλογος της ισχύος με την οποία μεταδίδει η ασύρματη διεπαφή και επομένως ο μεγαλύτερος ρυθμός προϋποθέτει μεγαλύτερη ισχύ εκπομπής.

Για τα πρωτόκολλα IEEE 802.11 b/g [16] υπάρχουν καθορισμένοι ρυθμοί μετάδοσης που υποστηρίζονται. Συγκεκριμένα οι ρυθμοί μετάδοσης που μπορούν να υποστηρίξουν οι διεπαφές είναι 1 Mbps, 2Mbps, 5.5 Mbps, 6 Mbps, 9 Mbps, 11 Mbps, 12 Mbps, 18Mbps, 24 Mbps, 36Mbps, 48 Mbps και 54 Mbps. Όπως θα δούμε στη συνέχεια κατά τη δημιουργία του slice ο ερευνητής επιλέγει ένα από τους παραπάνω ρυθμούς μετάδοσης για την ασύρματη διεπαφή που δεσμεύει.

- **Κανάλι Μετάδοσης (RadioChannel)**

Ένα ακόμα σημαντικό χαρακτηριστικό για την περιγραφή των ασύρματων διεπαφών είναι το κανάλι (συχνότητα) στο οποίο εκπέμπουν ή δέχονται πληροφορίες. Προκειμένου να επικοινωνήσουν δύο ασύρματες διεπαφές και να ανταλλάξουν πακέτα πληροφορίας θα πρέπει να λειτουργούν υπό το ίδιο κανάλι, δηλαδή υπό την ίδια συχνότητα. Σύμφωνα με τις προδιαγραφές των πρωτοκόλλων IEEE 802.11 b/g [16] που υποστηρίζουν τα μηχανήματα της εργαστηριακής πλατφόρμας υπάρχουν 13 διαθέσιμα ασύρματα κανάλια για μετάδοση πληροφορίας, τα οποία είναι νόμιμα στην Ελλάδα. Αυτά είναι τα εξής:

Wireless Channels for 802.11 b/g
IEEE 802.11b/g: Channel 1: 2412MHz
IEEE 802.11b/g: Channel 2: 2417MHz
IEEE 802.11b/g: Channel 3: 2422MHz
IEEE 802.11b/g: Channel 4: 2427MHz
IEEE 802.11b/g: Channel 5: 2432MHz
IEEE 802.11b/g: Channel 6: 2437MHz
IEEE 802.11b/g: Channel 7: 2442MHz
IEEE 802.11b/g: Channel 8: 2447MHz
IEEE 802.11b/g: Channel 9: 2452MHz
IEEE 802.11b/g: Channel 10: 2457MHz
IEEE 802.11b/g: Channel 11: 2462MHz
IEEE 802.11b/g: Channel 12: 2467MHz
IEEE 802.11b/g: Channel 13: 2472MHz

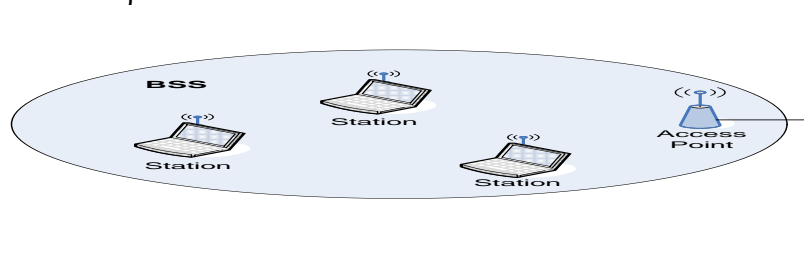
Ο ερευνητής όπως θα δούμε στην συνέχεια θα πρέπει να επιλέξει ένα από αυτά τα κανάλια για κάθε μια από τις ασύρματες επαφές που θα δεσμεύσει για το slice του.

- **Operating Mode**

Ακόμα ένα σημαντικό χαρακτηριστικό για την περιγραφή των interfaces είναι ο τρόπος λειτουργίας τους. Οι ασύρματες διεπαφές των κόμβων που διαθέτουμε μπορούν να λειτουργήσουν είτε ως Wireless Access Point (AP) είτε ως Wireless Client (STA). Ας εξετάσουμε αναλυτικότερα τι σημαίνουν αυτά τα δύο είδη λειτουργίας:

Wireless Access Point (AP)

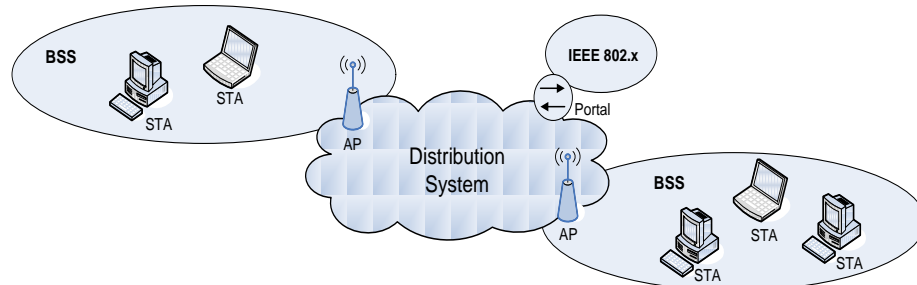
Wireless Access Points ([31]) είναι οι συσκευές που λειτουργούν ως κύριοι εκπομποί ή παραλήπτες σημάτων σε ένα ασύρματο δίκτυο. Το ασύρματο σημείο πρόσβασης (wireless access point) μπορεί να συνδεθεί με πολλούς τερματικούς σταθμούς (STA) σε μια σύνδεση ενός προς πολλαπλά σημεία (point to multipoint) και να δημιουργήσουν ένα τοπικό ασύρματο δίκτυο. Επιπλέον τα Access Points έχουν την δυνατότητα να συνδέσουν το ασύρματο δίκτυο που έχουν δημιουργήσει με ένα ενσύρματο Ethernet δίκτυο. Ένα σχήμα που αναπαριστά το ασύρματο τοπικό δίκτυο με ένα ασύρματο σημείο πρόσβασης και πολλαπλά τερματικά φαίνεται παρακάτω:



Σχήμα 3.4: Αναπαράσταση τοπικού ασύρματου δικτύου

Κεφάλαιο 3: Μοντέλα Περιγραφής Διαμοιραζόμενων Πόρων

Παράλληλα μπορούν να συνδεθούν δύο ή περισσότερα ασύρματα σημεία πρόσβασης (wireless access points) μεταξύ τους μέσω ενός ενσύρματου ή ασύρματου δικτύου (Distribution System) και να ενοποιήσουν τα δύο ασύρματα τοπικά δίκτυα ([31]). Ένα παράδειγμα σύνδεσης των δύο ασύρματων τοπικών δικτύων φαίνεται στο παρακάτω σχήμα:



Σχήμα 3.5: Διασύνδεση μεταξύ δύο ασυρμάτων σημείων πρόσβασης

Στην παρούσα διπλωματική κάθε μία από τις ασύρματες διεπαφές των ασύρματων κόμβων του εργαστηρίου μπορεί να χρησιμοποιηθεί ως AP στο ασύρματο δίκτυο που θα δημιουργήσει ο ερευνητής για το πείραμα του.

Wireless Client (STA)

Wireless Client (STA) ή αλλιώς κινητό τερματικό καλούμε μια συσκευή που υποστηρίζει την στοίβα πρωτοκόλλων IEEE 802.11 για ασύρματη διασύνδεση ([31]). Κάθε τερματικό μπορεί να λειτουργήσει είτε ως πομπός είτε ως παραλήπτης πακέτων πληροφορίας. Μπορεί να συνδεθεί και να ανταλλάξει πακέτα πληροφορίας μόνο με ένα άλλο κινητό τερματικό, το οποίο ανήκει στο ίδιο ασύρματο τοπικό δίκτυο και έχει το ίδιο κανάλι μετάδοσης (point to point connection). Αξίζει να επισημάνουμε ότι σύμφωνα με τις προδιαγραφές των IEEE 802.11 πρωτοκόλλων για τη λειτουργία των ασύρματων δικτύων όλα τα τερματικά θα πρέπει να συσχετιστούν με κάποιο Wireless Access Point πριν ξεκινήσουν να επικοινωνούν με άλλα στοιχεία του ασύρματου δικτύου. Στην παρούσα διπλωματική κάθε μία από τις ασύρματες διεπαφές των ασύρματων κόμβων του εργαστηρίου μπορεί να χρησιμοποιηθεί ως τερματικό (STA) στο ασύρματο δίκτυο που θα δημιουργήσει ο ερευνητής για το πείραμα του.

• **Wireless Mode**

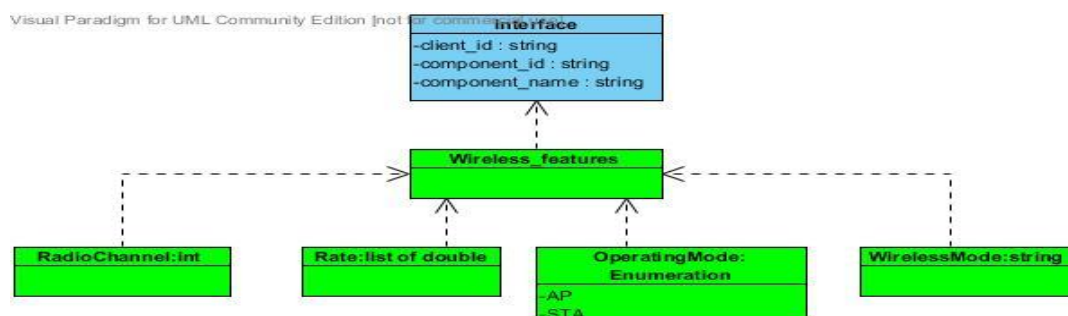
Ένα ακόμα σημαντικό χαρακτηριστικό είναι ο ασύρματος τρόπος λειτουργίας των διεπαφών που περιλαμβάνει τόσο το πρωτόκολλο της στοίβας πρωτοκόλλων IEEE 802.11 που χρησιμοποιείται όσο και το κανάλι λειτουργίας. Οι βασικοί τρόποι λειτουργίας σύμφωνα με τις προδιαγραφές του IEEE 802.11 είναι οι παρακάτω:

Mode	Number	Description
auto	0	Auto select operating mode
11a	1	802.11a (5GHz) mode (54Mbps)
11b	2	802.11b (2.4GHz) mode (11Mbps)
11g	3	802.11g (2.4GHz) mode with 802.11b compatibility (54Mbps)
fh	4	802.11 frequency hopping mode
11adt/111at	5	802.11a (5GHz) dynamic turbo mode
11gdt/11gt	6	802.11g (2GHz) dynamic turbo mode (108Mbps)
11ast	7	802.11a (5GHz) static turbo mode

Επειδή οι διεπαφές των κόμβων της πειραματικής μας πλατφόρμας δεν υποστηρίζουν το πρωτόκολλο IEEE 802.11a από τους παραπάνω τρόπους λειτουργίας μπορούμε να χρησιμοποιήσουμε μόνο όσους δεν εμπεριέχουν αυτό το πρωτόκολλο. Αξίζει να σημειώσουμε ότι στην παρούσα διπλωματική δεν χρησιμοποιήσαμε αυτό το χαρακτηριστικό για την κατάλληλη περιγραφή στο RSpec των ασύρματων διεπαφών, όμως μπορεί να χρησιμοποιηθεί σε μια μελλοντική επέκταση της παρούσας εργασίας και για αυτό το λόγο κρίναμε σωστό να το συμπεριλάβουμε στο σχήμα επέκτασης για την περιγραφή των ασύρματων κόμβων.

3.2.2 RSpec σχήμα επέκτασης

Το RSpec σχήμα επέκτασης που δημιουργήσαμε για να περιγράψουμε τις ασύρματες διεπαφές των ασύρματων κόμβων περιλαμβάνει ένα στοιχείο το `Wireless_features`, το οποίο έχει ως «παιδιά» τα χαρακτηριστικά που περιγράψαμε στο παραπάνω υποκεφάλαιο. Το στοιχείο αυτό αποτελεί επέκταση του στοιχείου `interface` που έχει οριστεί στο Core RSpec v.2 Schema [2]. Η επέκταση μπορεί να χρησιμοποιηθεί και στους τρεις τύπους RSpec v.2 για την περιγραφή ασύρματων πόρων. Το UML διάγραμμα που περιγράφει το extension για ασύρματους κόμβους είναι:



Σχήμα 3.6: UML διάγραμμα της επέκτασης για την περιγραφή ασύρματων κόμβων

Σε RNC αρχείο το extension είναι:

Κεφάλαιο 3: Μοντέλα Περιγραφής Διαμοιραζόμενων Πόρων

```
default namespace = "http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if"

InterfaceContents =
  element Wireless_features {
    element RadioChannel { xsd:int }* &
    element Rate list{{ xsd:double }*}* &
    element OperatingMode { "AP" | "STA"}* &
    element WirelessMode { xsd:string }*
  }*

start = InterfaceContents
```

Πίνακας 3.4: Σχήμα Επέκτασης σε rnc αρχείο

Όπως παρατηρούμε το σχήμα επέκτασης έχει διαφορετικό πεδίο ονομάτων από το Core RSpec v.2 schema και περιέχει τα χαρακτηριστικά που περιγράψαμε για τις ασύρματες διεπαφές. Επιπλέον βλέπουμε ότι δίνεται η δυνατότητα να δώσουμε μια λίστα ρυθμών μετάδοσης, το σύνολο των καναλιών μετάδοσης και των τρόπων λειτουργίας των διεπαφών. Αυτό χρησιμοποιείται στην περίπτωση του RSpec v.2 Advertisement όπου πρέπει να ενημερώσουμε το χρήστη για τους ρυθμούς μετάδοσης και τους τρόπους λειτουργίας που υποστηρίζονται από τις ασύρματες διεπαφές καθώς και για το σύνολο των διαθέσιμων καναλιών μετάδοσης για εκείνο το χρονικό διάστημα που θέλει.

Ας εξετάσουμε τώρα κάποια παραδείγματα των τριών τύπων RSpec v.2 όπως προέκυψαν στην παρούσα διπλωματική για να εξετάσουμε και τη χρήση του extension εντός του Core RSpec v.2.

RSpec v.2 Request

Παραθέτουμε ένα παράδειγμα RSpec v.2 Request στο οποίο ο ερευνητής θέλει να δεσμεύσει δύο κόμβους της εργαστηριακής πλατφόρμας και μία ασύρματη διεπαφή από κάθε κόμβο, όπου η μία θα λειτουργήσει ως Wireless Access Point (AP) και η άλλη ως Wireless Client (STA):

```
<?xml version="1.0"?>
<rspec
  xmlns="http://sorch.netmode.ntua.gr/ws/RSpec"
  xmlns:wif="http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://sorch.netmode.ntua.gr/ws/RSpec
http://sorch.netmode.ntua.gr/ws/RSpec/request.xsd
http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if
http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if/interfaces.xsd" type="request">
  <node
    client_id="omf.netmode.node2"
    component_id="urn:publicid:IDN+omf.netmode.ntua.gr+node+omf.netmode.node2"
    component_manager_id="urn:publicid:IDN+omf.netmode.ntua.gr+authority+cm" exclusive="true">
    <hardware_type name=" wireless access point "/>
    <sliver_type name="access point"/>
    <interface client_id="omf.netmode.node2:if0">
    <wif:Wireless_features>
    <wif:RadioChannel>3</wif:RadioChannel>
```

```

<wif:Rate>18</wif:Rate>
<wif:OperatingMode>STA</wif:OperatingMode>
</wif:Wireless_features>
</interface>
</node>
<node
client_id=" omf.netmode.node9"
component_id="urn:publicid:IDN+omf.netmode.ntua.gr+node+omf.netmode.node9"
component_manager_id="urn:publicid:IDN+omf.netmode.ntua.gr+authority+cm" exclusive="true">
<hardware_type name=" wireless access point "/>
<sliver_type name="access point"/>
<interface client_id=" omf.netmode.node9:if1">
<wif:Wireless_features>
<wif:RadioChannel>3</wif:RadioChannel>
<wif:Rate>54</wif:Rate>
<wif:OperatingMode>AP</wif:OperatingMode>
</wif:Wireless_features>
</interface>
</node>
</rspec>

```

Πίνακας 3.5: Παράδειγμα RSpec v.2 Request

Στο παραπάνω παράδειγμα ο ερευνητής ζητάει να δεσμεύσει δύο κόμβους που καθορίζονται από τα αντίστοιχα URNs στο πεδίο component_id του στοιχείου node. Αυτοί είναι οι κόμβοι 2 και 9 αφού στα αντίστοιχα URNs έχουν ονόματα (resource-name) omf.netmode.node2 και omf.netmode.node9. Επιπλέον με βάση το πεδίο client_id στο στοιχείο interface βλέπουμε ποιες διεπαφές των κόμβων θέλει να δεσμεύσει ο ερευνητής. Συγκεκριμένα στο παράδειγμα μας δεσμεύει τη διεπαφή if0 του κόμβου 2 και τη διεπαφή if1 του κόμβου 9. Παρατηρούμε ότι έχουν καθοριστεί και τα χαρακτηριστικά των ασύρματων διεπαφών με βάση το σχήμα επέκτασης που δημιουργήσαμε. Συγκεκριμένα το interface if0 του κόμβου 2 θα χρησιμοποιηθεί ως Wireless Client (STA), όπως φαίνεται στο element OperatingMode, θα λειτουργεί στο κανάλι IEEE 802.11b/g Channel 3:2422MHz, όπως φαίνεται από το element RadioChannel, και θα υποστηρίζει ρυθμό μετάδοσης 18 Mbps, όπως φαίνεται από το element Rate. Από την άλλη πλευρά το interface if1 του κόμβου 9 θα χρησιμοποιηθεί ως Wireless Access Point (AP), θα λειτουργεί στο κανάλι IEEE 802.11b/g Channel 3:2422MHz και θα υποστηρίζει ρυθμό μετάδοσης 54 Mbps.

RSpec v.2 Advertisement

Παραθέτουμε ένα παράδειγμα RSpec v.2 Advertisement το οποίο ενημερώνει τον ερευνητή ότι για το χρονικό διάστημα που θέλει να κάνει τα πειράματα του είναι διαθέσιμοι δύο κόμβοι της πειραματικής πλατφόρμας με κάποια χαρακτηριστικά, καθώς και δύο κανάλια μετάδοσης για κάθε μία από τις ασύρματες διεπαφές των κόμβων. Επιπλέον είναι διαθέσιμοι όλοι οι ρυθμοί μετάδοσης που υποστηρίζει η στοίβα πρωτοκόλλων IEEE 802.11 b/g καθώς και όλοι οι δυνατοί τρόποι λειτουργίας των ασύρματων διεπαφών:

```
<?xml version="1.0"?>
```

Κεφάλαιο 3: Μοντέλα Περιγραφής Διαμοιραζόμενων Πόρων

```

<rspec
xmlns="http://sorch.netmode.ntua.gr/ws/RSpec"
xmlns:wif="http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://sorch.netmode.ntua.gr/ws/RSpec
http://sorch.netmode.ntua.gr/ws/RSpec/ad.xsd
http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if
http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if/interfaces.xsd" type="advertisement">

<node
component_id="urn:publicid:IDN+omf.netmode.ntua.gr+node+omf.netmode.node2"
component_manager_id="urn:publicid:IDN+omf.netmode.ntua.gr+authority+cm"
component_name=" omf.netmode.node2" exclusive="true">
<available now="true"/>
<hardware_type name=" wireless access point "/>
<sliver_type name=" access point"/>
<interface
component_id="urn:publicid:IDN+omf.netmode.ntua.gr+interface+omf.netmode.node2:if0"
component_name=" omf.netmode.node2:if0">
<wif:Wireless_features>
<wif:Rate>1 2 5.5 6 9 11 12 18 24 36 48 54</wif:Rate>
<wif:RadioChannel>1</wif:RadioChannel>
<wif:RadioChannel>2</wif:RadioChannel>
<wif:OperatingMode>AP</wif:OperatingMode>
<wif:OperatingMode>STA</wif:OperatingMode>
</wif:Wireless_features>
</interface>
<interface
component_id="urn:publicid:IDN+omf.netmode.ntua.gr+interface+omf.netmode.node2:if1"
component_name=" omf.netmode.node2:if1">
<wif:Wireless_features>
<wif:Rate>1 2 5.5 6 9 11 12 18 24 36 48 54</wif:Rate>
<wif:RadioChannel>1</wif:RadioChannel>
<wif:RadioChannel>2</wif:RadioChannel>
<wif:OperatingMode>AP</wif:OperatingMode>
<wif:OperatingMode>STA</wif:OperatingMode>
</wif:Wireless_features>
</interface>
</node>

<node
component_id="urn:publicid:IDN+omf.netmode.ntua.gr+node+ omf.netmode.node9"
component_manager_id="urn:publicid:IDN+omf.netmode.ntua.gr+authority+cm"
component_name=" omf.netmode.node9" exclusive="true">
<available now="true"/>
<hardware_type name=" wireless access point "/>
<sliver_type name=" access point"/>
<interface
component_id="urn:publicid:IDN+omf.netmode.ntua.gr+interface+omf.netmode.node9:if0"
component_name=" omf.netmode.node9:if0">
<wif:Wireless_features>
<wif:Rate>1 2 5.5 6 9 11 12 18 24 36 48 54</wif:Rate>
<wif:RadioChannel>1</wif:RadioChannel>
<wif:RadioChannel>2</wif:RadioChannel>
<wif:OperatingMode>AP</wif:OperatingMode>
<wif:OperatingMode>STA</wif:OperatingMode>
</wif:Wireless_features>
</interface>
</node>

```

Κεφάλαιο 3: Μοντέλα Περιγραφής Διαμοιραζόμενων Πόρων

```
component_id="urn:publicid:IDN+omf.netmode.ntua.gr+interface+omf.netmode.node9:if1"
component_name="omf.netmode.node9:if1">
<wif:Wireless_features>
  <wif:Rate>1 2 5.5 6 9 11 12 18 24 36 48 54</wif:Rate>
  <wif:RadioChannel>1</wif:RadioChannel>
  <wif:RadioChannel>2</wif:RadioChannel>
  <wif:OperatingMode>AP</wif:OperatingMode>
  <wif:OperatingMode>STA</wif:OperatingMode>
</wif:Wireless_features>
</interface>
</node>
</rspec>
```

Πίνακας 3.6: Παράδειγμα RSpec v.2 Advertisement

Στο παραπάνω παράδειγμα το RSpec ενημερώνει τον ερευνητή ότι για το χρονικό διάστημα που θέλει να κάνει τα πειράματα του είναι διαθέσιμοι δύο κόμβοι οι omf.netmode.node2 και omf.netmode.node9, όπως φαίνεται από τα αντίστοιχα URNs. Επιπλέον κάθε ένας από τους παραπάνω κόμβους έχει δύο διεπαφές διαθέσιμες, οι οποίες καθορίζονται από τα αντίστοιχα URNs των interfaces στο υποχρεωτικό πεδίο component_id του στοιχείου interface. Ακόμα βλέπουμε ότι γίνεται χρήση του extension schema για να καθοριστούν τα κανάλια μετάδοσης που είναι διαθέσιμα εκείνο το χρονικό διάστημα και δεν έχουν εκχωρηθεί σε διεπαφές κάποιου άλλου slice, καθώς και οι ρυθμοί μετάδοσης και οι τρόποι λειτουργίας που είναι διαθέσιμοι. Μπορούμε να επιλέξουμε ένα διαθέσιμο ρυθμό μετάδοσης, κανάλι μετάδοσης και τρόπο λειτουργίας για κάθε ένα από τα διαθέσιμα interfaces των κόμβων. Συγκεκριμένα στο παράδειγμά μας παρατηρούμε ότι στο χρονικό διάστημα που θέλουμε διαθέσιμα είναι τα κανάλια IEEE 802.11 b/g 1 και 2, οι ρυθμοί μετάδοσης 1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48 και 54 Mbps και οι τρόποι λειτουργίας της διεπαφής ως Wireless Access Point (AP) ή Wireless Client (STA).

RSpec v.2 Manifest

Παραθέτουμε ένα παράδειγμα RSpec v.2 Manifest το οποίο ενημερώνει τους χρήστες ότι κάποιο slice ενός ερευνητή περιέχει δύο κόμβους και ένα interface από κάθε έναν από αυτούς. Μάλιστα όπως θα δούμε στο παρακάτω παράδειγμα η μία ασύρματη διεπαφή θα λειτουργήσει ως Wireless Access Point (AP) και η άλλη ως Wireless Client(STA):

```
<?xml version="1.0"?>

<rspec
xmlns="http://sorch.netmode.ntua.gr/ws/RSpec"
xmlns:wif="http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://sorch.netmode.ntua.gr/ws/RSpec
http://sorch.netmode.ntua.gr/ws/RSpec/manifest.xsd
http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if
http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if/interfaces.xsd" type="manifest">

<node
client_id="omf.netmode.node3"
```

Κεφάλαιο 3: Μοντέλα Περιγραφής Διαμοιραζόμενων Πόρων

```
component_id="urn:publicid:IDN+omf.netmode.ntua.gr+node+omf.netmode.node3"
component_manager_id="urn:publicid:IDN+omf.netmode.ntua.gr+authority+cm"
sliver_id="urn:publicid:IDN+omf.netmode.ntua.gr+sliver+omf.netmode.node3" exclusive="true">
<hardware_type name=" wireless access point "/>
<sliver_type name="access point"/>
<interface
component_id="urn:publicid:IDN+omf.netmode.ntua.gr+interface+omf.netmode.node3:if0"
client_id="omf.netmode.node3:if0">
<wif:Wireless_features>
<wif:Rate>18</wif:Rate>
<wif:RadioChannel>1</wif:RadioChannel>
<wif:OperatingMode>STA</wif:OperatingMode>
</wif:Wireless_features>
</interface>
</node>

<node
client_id="omf.netmode.node4"
component_id="urn:publicid:IDN+omf.netmode.ntua.gr+node+omf.netmode.node4"
component_manager_id="urn:publicid:IDN+omf.netmode.ntua.gr+authority+cm"
sliver_id="urn:publicid:IDN+omf.netmode.ntua.gr+sliver+omf.netmode.node4" exclusive="true">
<hardware_type name=" wireless access point "/>
<sliver_type name="access point"/>
<interface
component_id="urn:publicid:IDN+omf.netmode.ntua.gr+interface+omf.netmode.node4:if1"
client_id="omf.netmode.node4:if1">
<wif:Wireless_features>
<wif:Rate>18</wif:Rate>
<wif:RadioChannel>1</wif:RadioChannel>
<wif:OperatingMode>AP</wif:OperatingMode>
</wif:Wireless_features>
</interface>
</node>

</rspec>
```

Πίνακας 3.7: Παράδειγμα RSpec v.2 Manifest

Στο παραπάνω παράδειγμα παρατηρούμε ότι έχουν εκχωρηθεί δύο κόμβοι στο slice του ερευνητή, οι οποίοι καθορίζονται από τα αντίστοιχα URNs στο attribute component_id του στοιχείου node. Αυτοί είναι οι κόμβοι 3 και 4, αφού στα αντίστοιχα URNs έχουν ονόματα (resource-name) omf.netmode.node3 και omf.netmode.node4. Στο πεδίο sliver_id έχουμε δώσει το URN του κάθε ασύρματου κόμβου, αφού ο κάθε χρήστης χρησιμοποιεί κατα αποκλειστικό τρόπο τον κόμβο εφόσον δεν έχουμε χωρίσει το κόμβο σε slivers. Επιπλέον με βάση τα URNs των διεπαφών στο πεδίο component_id του στοιχείου interface βλέπουμε ποιες διεπαφές των κόμβων έχει εκχωρήσει ο ερευνητής στο δεδομένο slice. Συγκεκριμένα στο παράδειγμα μας έχει εκχωρήσει τη διεπαφή if0 του κόμβου 3 και τη διεπαφή if1 του κόμβου 4. Παρατηρούμε ότι έχουν καθοριστεί και τα χαρακτηριστικά των ασύρματων διεπαφών με βάση το extension schema που δημιουργήσαμε. Συγκεκριμένα το interface if0 του κόμβου 3 χρησιμοποιείται ως Wireless Client (STA), όπως φαίνεται στο element OperatingMode, λειτουργεί στο κανάλι IEEE 802.11b/g Channel 1:2412MHz, όπως φαίνεται από το element RadioChannel, και υποστηρίζει ρυθμό μετάδοσης 18 Mbps, όπως φαίνεται από το element Rate. Από την

άλλη πλευρά το interface if1 του κόμβου 4 χρησιμοποιείται ως Wireless Access Point (AP), λειτουργεί στο κανάλι IEEE 802.11b/g Channel 1:2412MHz και υποστηρίζει ρυθμό μετάδοσης 18 Mbps.

3.3 RSpec έλεγχος εγκυρότητας

Όπως αναφέραμε και προηγουμένως το RSpec θα πρέπει να συμμορφώνεται με βάση κάποια σχήματα κορμού ανάλογα με το είδος στο οποίο ανήκει έτσι ώστε να επιτραπεί η επεξεργασία των πληροφοριών που περιέχει. Επιπλέον επειδή το RSpec παρέχει τη δυνατότητα να δημιουργήσουμε επεκτάσεις, οι οποίες θα έχουν διαφορετικό πεδίο ονομάτων από το Core RSpec, δημιουργείται η ανάγκη για έλεγχο εγκυρότητας με δυο ή περισσότερα σχήματα (Core RSpec και extension schemas). Αυτό επιτυγχάνεται με χρήση της γλώσσας NVDL που αναλύσαμε παραπάνω και διάφορων εργαλείων που έχουν αναπτυχθεί σε γλώσσες προγραμματισμού για την εφαρμογή αυτής της γλώσσας σε XML κείμενα, έτσι ώστε να ελέγξουμε την εγκυρότητα τους.

Ένα από τα εργαλεία που έχουν αναπτυχθεί για χρήση της γλώσσας NVDL για τον έλεγχο της εγκυρότητας σε XML κείμενα που περιέχουν δύο ή περισσότερα πεδία ονομάτων είναι το JNVDL [32]. Το JNVDL είναι ένα εργαλείο υλοποιημένο στη γλώσσα προγραμματισμού JAVA, το οποίο χρησιμοποιήσαμε για να ελέγξουμε αν είναι έγκυρα τα RSpec αρχεία που ανταλλάσσονται μεταξύ πελάτη και εξυπηρετητή.

Συγκεκριμένα παίρνει ως είσοδο το nvdI αρχείο με τους κανόνες που αφορούν το validation των διάφορων στοιχείων τους με βάση το πεδίο ονομάτων τους και το XML αρχείο στο οποίο θέλουμε να κάνουμε έλεγχο εγκυρότητας.

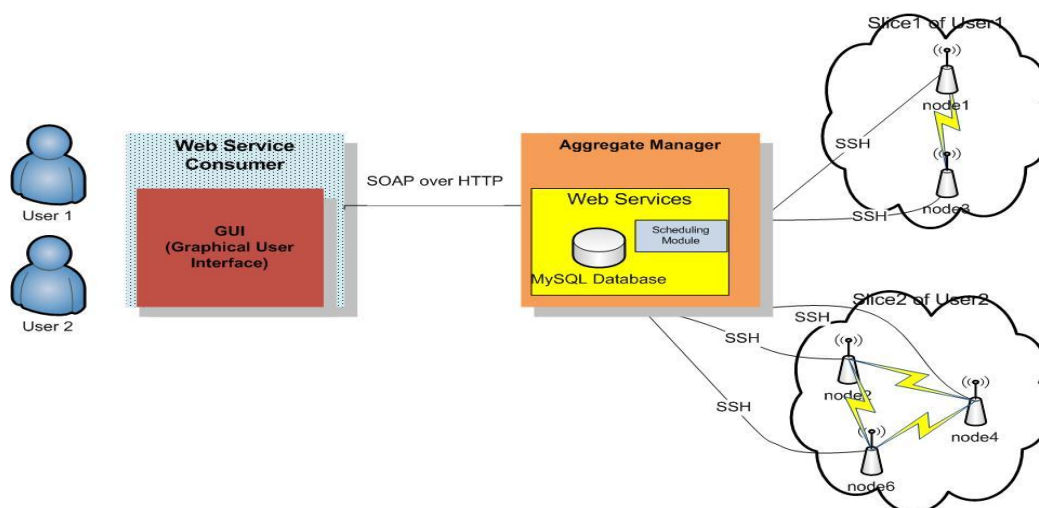
Κεφάλαιο 4

Αρχιτεκτονική του Συστήματος

Με χρήση της δομής δεδομένων RSpec v.2 και της επέκτασης που δημιουργήσαμε ώστε να περιγράψουμε κατάλληλα ασύρματους κόμβους σχεδιάσαμε ένα σύστημα με το οποίο ο χρήστης θα μπορεί να δημιουργεί καινούργια slices και να διαχειρίζεται τα slices που έχει ήδη δημιουργήσει. Το σύστημα αυτό βασίζεται στον σχεδιασμό κατάλληλων υπηρεσιών διαδικτύου (web services) που επιτελούν τις διάφορες λειτουργίες διαχείρισης. Οι υπηρεσίες αυτές επικοινωνούν με μια βάση δεδομένων όπου υπάρχουν οι καταγραφές για όλα τα slices των χρηστών και παρέχουν τις απαιτούμενες πληροφορίες για τη διαχείρισή τους. Παράλληλα αναπτύξαμε ένα εργαλείο χρονοπρογραμματισμού σε κάποιες από τις υπηρεσίες διαδικτύου έτσι ώστε να δημοσιοποιούνται στο χρήστη μόνο οι διαθέσιμοι πόροι της πειραματικής πλατφόρμας όταν θέλει να δημιουργήσει κάποιο slice για κάποιο συγκεκριμένο χρονικό διάστημα. Αυτό έχει ως στόχο να μην υπάρχει χρονική επικάλυψη χρήσης των ίδιων πόρων της πειραματικής πλατφόρμας από δύο ή περισσότερα slices. Αξίζει να σημειώσουμε πως οι διαμοιραζόμενοι πόροι της πειραματικής πλατφόρμας στο πεδίο του χρόνου είναι οι ασύρματοι κόμβοι καθώς και το σύνολο των καναλιών μετάδοσης που έχουμε ορίσει στις διεπαφές κάποιου slice. Συνεπώς με την ανάπτυξη αυτού του συστήματος οι διάφοροι χρήστες θα μπορούν να δημιουργήσουν και να διαχειριστούν slices που τους ανήκουν.

4.1 Περιγραφή της Αρχιτεκτονικής

Η αρχιτεκτονική του συστήματος που αναπτύξαμε παρουσιάζεται σχηματικά στο παρακάτω σχήμα:



Σχήμα 4.1: Αναπαράσταση της αρχιτεκτονικής του συστήματος

Κεφάλαιο 4: Αρχιτεκτονική Συστήματος

Το σύστημα που σχεδιάσαμε περιλαμβάνει στην πλευρά του πελάτη έναν Web Service Consumer και στην πλευρά του εξυπηρετητή έναν Aggregate Manager. Ας εξετάσουμε αναλυτικά τι περιλαμβάνουν οι δύο πλευρές:

Web Service Consumer

Στην πλευρά του πελάτη έχει σχεδιαστεί κατάλληλο γραφικό περιβάλλον χρήστη μέσω του οποίου οι χρήστες μπορούν να χρησιμοποιήσουν τις διάφορες υπηρεσίες διαδικτύου. Στο γραφικό περιβάλλον μπορούν να έχουν πρόσβαση περισσότεροι από έναν χρήστες (ερευνητές) εφόσον διαθέτουν έγκυρο συνδυασμό username/password. Μέσω του γραφικού περιβάλλοντος οι διάφοροι χρήστες μπορούν να δημιουργήσουν κάποιο καινούργιο slice με τους διαθέσιμους πόρους, το οποίο θα ενεργοποιήσουν για κάποιο χρονικό διάστημα προκειμένου να πραγματοποιήσουν τα πειράματά τους, καθώς και να διαχειριστούν τα ήδη υπάρχοντα slices που έχουν δημιουργήσει. Η επικοινωνία με τις διάφορες υπηρεσίες διαδικτύου γίνεται με SOAP μηνύματα πάνω από HTTP πρωτόκολλο μεταφοράς [33] (SOAP over HTTP).

Aggregate Manager

Στην πλευρά του εξυπηρετητή έχουμε υλοποιήσει έναν aggregate manager, ο οποίος περιλαμβάνει τις υπηρεσίες διαδικτύου (web services) για τη δημιουργία και τη διαχείριση των slices από τους διάφορους ερευνητές. Οι υπηρεσίες διαδικτύου που έχουμε σχεδιάσει επιτελούν λειτουργίες για τη διαχείριση των slices του κάθε χρήστη, όπως δημιουργία καινούργιου slice με τους διαθέσιμους πόρους, τροποποίηση ήδη υπάρχοντος, κράτηση ενός slice, απενεργοποίηση ενός ενεργοποιημένου slice, ακύρωση κρατήσεων και διαγραφή κάποιων slices. Η λειτουργία των υπηρεσιών βασίζεται στις πληροφορίες μιας βάσης δεδομένων MySQL [34], η οποία περιέχει καταγραφές για τα slices των χρηστών και ποιοί πόροι έχουν εκχωρηθεί σε αυτά καθώς και για τις κρατήσεις τους.

Παράλληλα έχουμε αναπτύξει ένα εργαλείο χρονοπρογραμματισμού, παρόμοιο με αυτό που περιγράφεται στη δημοσίευση [3], σε κάποιες λειτουργίες των υπηρεσιών ώστε να μην υπάρχει χρονική επικάλυψη χρήσης πόρων (ασύρματων κόμβων και καναλιών μετάδοσης) από διαφορετικά slices. Η δημιουργία αυτού του εργαλείου είναι πολύ σημαντική για τη δημοσιοποίηση των διαθέσιμων ασύρματων πόρων για κάποιο δεδομένο χρονικό διάστημα. Με τη βοήθεια αυτού του εργαλείου μπορούν οι διάφοροι χρήστες να χρησιμοποιούν τους πόρους της πειραματικής πλατφόρμας χωρίς να υπάρχει επικάλυψη. Η λειτουργία του εργαλείου βασίζεται στις πληροφορίες της βάσης δεδομένων σχετικά με τις κρατήσεις (reservations) που έχουν γίνει για τα διάφορα slices.

Το χρονικό διάστημα που το slice είναι ενεργοποιημένο (activated) ο χρήστης έχει πρόσβαση στους διάφορους κόμβους που έχουν εκχωρηθεί στο slice. Συγκεκριμένα έχει secure shell access (SSH) [35] και μπορεί να πραγματοποιήσει τα πειράματά του, χρησιμοποιώντας όμως και τα αντίστοιχα χαρακτηριστικά που έχει

ορίσει για κάθε ασύρματη διεπαφή, τα οποία είναι ρυθμός μετάδοσης, κανάλι μετάδοσης και τρόπος λειτουργίας.

4.2 Περιγραφή της Βάσης Δεδομένων

Για τη λειτουργία του συστήματος που δημιουργήσαμε ήταν απαραίτητη η δημιουργία μιας βάσης δεδομένων, που θα περιλαμβάνει κατάλληλους πίνακες (tables) με τις απαραίτητες πληροφορίες για την ορθή λειτουργία του. Για αυτό το λόγο επεκτείναμε την ήδη υπάρχουσα βάση δεδομένων MySQL inventory2 του OMF, η οποία περιέχει πληροφορίες για τους ασύρματους κόμβους της πειραματικής πλατφόρμας του εργαστηρίου Διαχείρισης και Σχεδιασμού Δικτύων Τηλεματικής. Στην υπάρχουσα βάση δεδομένων προσθέσαμε κάποιους πίνακες έτσι ώστε να υποστηριχθεί η αρχιτεκτονική με χρησιμοποίηση της έννοιας των slices. Μέσω του συστήματος θα πρέπει οι διάφοροι χρήστες να μπορούν να διαχειρίζονται τα slices που έχουν δημιουργήσει. Αυτοί οι πίνακες είναι οι slices, slice_node_association και node_if_association οι οποίοι περιέχουν πληροφορίες για τα διάφορα slices καθώς και για τους πόρους που έχουν εκχωρηθεί σε αυτά. Επιπλέον ήταν αναγκαίο να φτιάξουμε ένα πίνακα (login) ο οποίος θα διατηρεί στοιχεία ταυτοποίησης (username/password) των διάφορων χρηστών. Τέλος, προκειμένου να υλοποιήσουμε τον χρονοπρογραμματιστή εισαγάγαμε τον πίνακα reservations, ο οποίος περιέχει όλες τις κρατήσεις των slices που έχουν γίνει από τους διάφορους χρήστες. Με τις πληροφορίες που μας παρέχει ο πίνακας reservations μπορούμε να γνωρίζουμε τους διαθέσιμους πόρους για κάθε χρονικό διάστημα και να κάνουμε time division των ασύρματων κόμβων και των καναλιών μετάδοσης, αφού ελέγχουμε κάθε φορά να μην υπάρχει χρονική επικάλυψη χρήσης ασύρματων κόμβων και καναλιών μετάδοσης από διαφορετικά slices.

Ας εξετάσουμε αναλυτικά τι περιέχει κάθε ένας από αυτούς τους πίνακες καθώς και ο πίνακας nodes που υπήρχε στην βάση δεδομένων inventory2 και περιέχει πληροφορίες για τους ασύρματους κόμβους.

Πίνακας nodes

Ο πίνακας nodes περιέχει πληροφορίες για τα χαρακτηριστικά των ασύρματων κόμβων. Αναλυτικά περιέχει τα παρακάτω πεδία:

<i>id</i>	Αποτελεί το κλειδί (Primary key) του πίνακα αυτού και είναι ο αριθμός που καθορίζει σε ποιόν ασύρματο κόμβο της πειραματικής πλατφόρμας του εργαστηρίου αναφερόμαστε. Θα το χρησιμοποιήσουμε και σε κάποιους από τους επόμενους πίνακες προκειμένου να συσχετίσουμε τους δύο πίνακες.
<i>control_ip</i>	Το πεδίο αυτό περιέχει την IP διεύθυνση που έχει ορίσει το OMF σε κάθε ασύρματο κόμβο για λόγους διαχείρισης.
<i>control_mac</i>	Το πεδίο αυτό περιέχει τη διεύθυνση MAC που

Κεφάλαιο 4: Αρχιτεκτονική Συστήματος

	έχει οριστεί σε κάθε ασύρματο κόμβο για λόγους διαχείρισης.
<i>hostname</i>	Το πεδίο αυτό περιέχει το όνομα του ασύρματου κόμβου το οποίο το διαχωρίζει από τους υπόλοιπους κόμβους της πειραματικής πλατφόρμας.
<i>hrn (human readable name)</i>	Είναι ένα όνομα που έχει δώσει σε κάθε κόμβο το OMF προκειμένου να εκτελέσει κάποιες εσωτερικές λειτουργίες του. Για παράδειγμα το hrn του node1 είναι omf.netmode.node1. Το hrn χρησιμοποιείται και σαν όνομα στα αντίστοιχα URNs που περιγράφουν τον κόμβο προκειμένου να τον διαχωρίζουν από τους άλλους κόμβους.

Πίνακας 4.1: Πίνακας nodes

Πίνακας login

Ο πίνακας login είναι απαραίτητος για την ταυτοποίηση των χρηστών της πειραματικής πλατφόρμας, αφού περιέχει τους έγκυρους συνδυασμούς username/password. Εφόσον ο χρήστης δώσει το σωστό συνδυασμό username/password το σύστημα ελέγχοντας αυτό τον πίνακα του δίνει πρόσβαση στο γραφικό περιβάλλον ώστε να μπορέσει να δημιουργήσει κάποιο slice ή να διαχειριστεί τα slices που έχει ήδη δημιουργήσει. Αναλυτικά περιέχει τα παρακάτω πεδία:

<i>username</i>	Αποτελεί το κλειδί (Primary key) του πίνακα αυτού και είναι το όνομα του συγκεκριμένου χρήστη. Το όνομα αυτό περνάει μέσω php \$_SESSION απο την σελίδα του authentication στις επόμενες σελίδες ώστε να μπορεί ο χρήστης να διαχειριστεί μόνο δικά του slices και να μην επηρεάζει slices των άλλων χρηστών που χρησιμοποιούν την πειραματική πλατφόρμα.
<i>password</i>	Είναι ο κωδικός που έχει ο κάθε χρήστης ώστε να αποκτήσει πρόσβαση στο γραφικό περιβάλλον. Μόνο αν ο χρήστης δώσει το σωστό συνδυασμό username/password έχει πρόσβαση στο γραφικό περιβάλλον.

Πίνακας 4.2: Πίνακας login

Πίνακας slices

Ο πίνακας slices είναι ένας πίνακας που κατασκευάσαμε για να υποστηρίξουμε την αρχιτεκτονική με χρήση slices από τους χρήστες στην οποία βασίζεται το σύστημα μας. Περιέχει πληροφορίες για τα διάφορα slices ώστε να είναι εύκολα διαχειρίσιμα από τους διάφορους χρήστες. Αναλυτικά περιέχει τα παρακάτω πεδία :

Κεφάλαιο 4: Αρχιτεκτονική Συστήματος

<i>slice_id</i>	Αποτελεί το κλειδί (Primary key) του πίνακα αυτού και είναι ο αριθμός που καθορίζει σε ποιά slice της πειραματικής πλατφόρμας αναφερόμαστε. Χρησιμοποιείτε και σε κάποιους απο τους επόμενους πίνακες προκειμένου να συσχετίσουμε τους πίνακες με τον πίνακα slices.
<i>name</i>	Είναι το όνομα που έχει δώσει ο χρήστης στο slice που έχει δημιουργήσει και είναι μοναδικό. Δεν μπορεί να υπάρχουν δύο ή περισσότερα slices που να έχουν το ίδιο όνομα.
<i>owner</i>	Στο πεδίο αυτό αναφέρεται ο χρήστης που έχει δημιουργήσει το συγκεκριμένο slice και ο οποίος έχει δικαίωμα να το χρησιμοποιήσει για τα πειράματα του και να το διαχειριστεί. Όπως αναφέραμε και προηγουμένως κάθε slice που έχει δημιουργηθεί στην πειραματική πλατφόρμα για κάποιο συγκεκριμένο πείραμα έχει ένα χρήστη, ο οποίος είναι αυτός που το δημιούργησε.
<i>number_nodes</i>	Ο αριθμός των ασύρματων κόμβων που έχουν εκχωρηθεί σε ένα συγκεκριμένο slice.
<i>activate</i>	Είναι μια σημαία (flag) που δείχνει αν το συγκεκριμένο slice είναι ενεργοποιημένο αυτή τη χρονική στιγμή ή όχι. Αν η σημαία είναι ίση με 1 τότε το slice είναι ενεργοποιημένο ενώ αν είναι ίση με 0 τότε δεν είναι ενεργοποιημένο. Το πεδίο αυτό είναι πολύ σημαντικό στην υλοποίηση του συστήματος αφού μας επιτρέπει να γνωρίζουμε αν το slice είναι ενεργοποιημένο αυτή τη χρονική στιγμή και έτσι μπορούμε να δώσουμε στον χρήστη του πρόσβαση στους πόρους που έχουν εκχωρηθεί στο slice. Αυτό μπορεί να υλοποιηθεί με ένα shell script το οποίο θα επικοινωνεί με τη βάση δεδομένων και θα ελέγχει τη σημαία αυτή. Αν η σημαία είναι 1 τότε θα εκτελεί τις απαραίτητες λειτουργίες ώστε να δώσει στο χρήστη του αντίστοιχου slice SSH (secure shell) πρόσβαση στους ασύρματους κόμβους.
<i>timeactivation</i>	Είναι η χρονική στιγμή έναρξης της ενεργοποίησης του συγκεκριμένου slice. Στο πεδίο αυτό αναγράφεται μια συγκεκριμένη χρονική στιγμή εφόσον το slice είναι ενεργοποιημένο αυτή τη χρονική στιγμή, δηλαδή το activate έχει τιμή ίση με 1 και ενημερώνει για το πότε ξεκίνησε η ενεργοποίησή του.
<i>duration</i>	Είναι η χρονική διάρκεια ενεργοποίησης του συγκεκριμένου slice σε ώρες. Στο πεδίο αυτό

Κεφάλαιο 4: Αρχιτεκτονική Συστήματος

	αναγράφεται μια χρονική διάρκεια σε ώρες εφόσον το slice είναι ενεργοποιημένο αυτή τη χρονική στιγμή, δηλαδή το activate έχει τιμή ίση με 1 και ενημερώνει σχετικά με τη χρονική διάρκεια ενεργοποίησης του.
<i>timemodification</i>	Η χρονική στιγμή όπου συντελέστηκε κάποια αλλαγή στην καταχώριση του πίνακα για το συγκεκριμένο slice.

Πίνακας 4.3: Πίνακας slices

Πίνακας slice_node_association

Ο πίνακας slice_node_association δημιουργήθηκε για να συσχετίσει τους διάφορους ασύρματους κόμβους με τα slices, ώστε να γνωρίζουμε ποιό ασύρματος κόμβος έχει εκχωρηθεί στα διάφορα slices. Αναλυτικά περιέχει τα παρακάτω πεδία:

<i>slice_id</i>	Είναι το id του συγκεκριμένου slice στο οποίο αναφερόμαστε.
<i>node_id</i>	Είναι το id του ασύρματος κόμβου που έχει εκχωρηθεί στο συγκεκριμένο slice.

Πίνακας 4.4: Πίνακας slice_node_association

Αξίζει να επισημάνουμε ότι ο κάθε συνδυασμός slice_id και node_id είναι μοναδικός και αποτελεί το Primary Key του πίνακα slice_node_association. Κάθε καταχώριση του πίνακα δηλώνει ότι ο κόμβος με id ίσο με node_id έχει εκχωρηθεί στο slice με id ίσο με slice_id.

Πίνακας node_if_association

Ο πίνακας αυτός περιέχει τις πληροφορίες σχετικά με το ποιές διεπαφές έχουν ενεργοποιηθεί σε κάθε ασύρματο κόμβο που έχει εκχωρηθεί σε ένα slice και με ποία χαρακτηριστικά. Να υπενθυμίσουμε ότι κάθε ασύρματος κόμβος έχει δύο διαθέσιμες ασύρματες διεπαφές στις οποίες ορίζουμε κάποια χαρακτηριστικά, τα οποία είναι ο ρυθμός μετάδοσης, το κανάλι μετάδοσης και ο τρόπος λειτουργίας τους (AP ή STA). Αναλυτικά περιέχει τα παρακάτω πεδία :

<i>slice_id</i>	Το id του slice στο οποίο έχει εκχωρηθεί ο συγκεκριμένος κόμβος και η αντίστοιχη διεπαφή του.
<i>node_id</i>	Το id του ασύρματος κόμβου που έχει εκχωρηθεί στο συγκεκριμένο slice με id ίσο με slice_id.
<i>if_id</i>	Το id της ασύρματης διεπαφής του κόμβου με id ίσο με node_id το οποίο έχει ενεργοποιηθεί στο slice με id ίσο με slice_id. Αν αναφερόμαστε στην πρώτη ασύρματη διεπαφή του κόμβου τότε το if_id είναι if0, ενώ αν αναφερόμαστε στην δεύτερη το id είναι if1.

Κεφάλαιο 4: Αρχιτεκτονική Συστήματος

<i>Rate</i>	Ο ρυθμός μετάδοσης σε Mbps που έχει ορίσει ο χρήστης του slice για την συγκεκριμένη ασύρματη διεπαφή.
<i>RadioChannel</i>	Το κανάλι μετάδοσης (συγκεκριμένα ο αριθμός του καναλιού) που έχει ορίσει ο χρήστης του slice για την συγκεκριμένη ασύρματη διεπαφή.
<i>OperatingMode</i>	Ο τρόπος λειτουργίας (AP ή STA) που έχει ορίσει ο χρήστης του slice για την συγκεκριμένη ασύρματη διεπαφή.

Πίνακας 4.5: Πίνακας node_if_association

Αξίζει να επισημάνουμε ότι ο κάθε συνδυασμός slice_id, node_id και if_id είναι μοναδικός και αποτελεί το Primary Key του πίνακα node_if_association. Κάθε καταχώριση του πίνακα δηλώνει ότι στο slice με id ίσο με slice_id έχει εκχωρηθεί η ασύρματη διεπαφή με id ίσο με if_id του ασύρματου κόμβου με id ίσο με node_id (ο οποίος έχει και αυτός εκχωρηθεί στο συγκεκριμένο slice) με τα αντίστοιχα χαρακτηριστικά Rate, RadioChannel και OperatingMode.

Πίνακας reservations

Ο πίνακας reservations περιέχει όλες τις κρατήσεις (reservations) των slices που έχουν γίνει από τους διάφορους χρήστες που έχουν δημιουργήσει τα slices. Οι πληροφορίες που περιέχει αυτός ο πίνακας είναι πολύ σημαντικές ώστε να υλοποιήσουμε τη λειτουργία χρονοπρογραμματισμού σε διάφορες λειτουργίες των υπηρεσιών διαδικτύου και να αποφύγουμε τη χρονική επικάλυψη χρήσης των ίδιων κόμβων από διαφορετικά slices καθώς και των ίδιων καναλιών από διεπαφές που έχουν εκχωρηθεί σε διαφορετικά slices. Συνεπώς με χρήση των πληροφοριών του πίνακα αυτού ο χρήστης μπορεί να επιλέξει μόνο τους διαθέσιμους πόρους για κάποιο χρονικό διάστημα που θέλει να πραγματοποιήσει το πείραμα του. Αναλυτικά περιέχει τα παρακάτω πεδία:

name	Το όνομα του slice στο οποίο έχει γίνει η συγκεκριμένη κράτηση.
time_activation_start	Η χρονική στιγμή έναρξης της ενεργοποίησης του συγκεκριμένου slice με όνομα name.
time_activation_end	Η χρονική στιγμή λήξης της ενεργοποίησης του συγκεκριμένου slice με όνομα name.

Πίνακας 4.6: Πίνακας reservations

Αξίζει να επισημάνουμε ότι ο κάθε συνδυασμός name, time_activation_start και time_activation_end είναι μοναδικός και αποτελεί το Primary Key του πίνακα reservations. Ο πίνακας μπορεί να περιέχει πολλαπλές κρατήσεις του ίδιου slice για κάποια χρονικά διαστήματα.

Κεφάλαιο 5

Περιγραφή Υπηρεσιών Ιστού

Βασιζόμενοι στο RSpec v.2 που επεκτείναμε για την περιγραφή ασύρματων πόρων σχεδιάσαμε κάποιες υπηρεσίες ιστού (web services) σε γλώσσα php [36] για τη δημοσίευση των διαθέσιμων πόρων και τη διαχείριση τους από τον ερευνητή. Παράλληλα αναπτύξαμε ένα εργαλείο χρονοπρογραμματισμού για τους διαμοιραζόμενους πόρους της πειραματικής πλατφόρμας. Οι υπηρεσίες αυτές αποτέλεσαν τη βάση για τη λειτουργία του συστήματος που δημιουργήσαμε. Η λογική που ακολουθήσαμε είναι βασισμένη στη δημιουργία και διαχείριση slices από τους ερευνητές που χρησιμοποιούν την πειραματική πλατφόρμα. Τα slices των ερευνητών θα περιέχουν διαμοιραζόμενους πόρους της πειραματικής πλατφόρμας. Οι πόροι που διαμοιράζουμε χρονικά (time division) στα διάφορα slices των ερευνητών ώστε να πετύχουμε πιο αξιόπιστη χρήση της πειραματικής πλατφόρμας είναι οι ασύρματοι κόμβοι και τα κανάλια μετάδοσης στα οποία εκπέμπουν ή δέχονται πακέτα πληροφορίας οι ασύρματες διεπαφές που ανήκουν στο slice.

Με τον σχεδιασμό των υπηρεσιών διαδικτύου οι διάφοροι ερευνητές που χρησιμοποιούν τους πόρους της πειραματικής μας πλατφόρμας θα έχουν την δυνατότητα να δημιουργήσουν ένα ή περισσότερα slices με πόρους που είναι διαθέσιμοι στο χρονικό διάστημα που επιθυμούν, να τα τροποποιήσουν προσθέτοντας ή αφαιρώντας κάποιους πόρους αν είναι εφικτό και να τα διαγράψουν.

Για τη δημιουργία των υπηρεσιών ιστού χρησιμοποιήσαμε τόσο το WSO2 wsf extension για php web services [30] και τη λειτουργία του χωρίς WSDL αρχείο (NO-WSDL mode), όσο και τις συναρτήσεις της php για τη δημιουργία php υπηρεσιών διαδικτύου [36] με τη χρήση κατάλληλων wsdl αρχείων για την περιγραφή των υπηρεσιών αυτών.

5.1 Βασικές Έννοιες και Παραδοχές

5.1.1 Καταστάσεις των slices

Για τον σχεδιασμό των υπηρεσιών εισάγαμε κάποιες έννοιες σχετικά με την κατάσταση των διάφορων slices την παρούσα χρονική στιγμή. Καταρχήν να σημειώσουμε ότι με την έννοια **reserved slices** ορίζουμε το σύνολο των slices που έχουν δεσμευθεί την παρούσα χρονική στιγμή για ένα ή περισσότερα χρονικά διαστήματα στο μέλλον και με την έννοια **no-reserved slices** ορίζουμε το σύνολο των slices που δεν έχουν δεσμευθεί την παρούσα χρονική στιγμή για μελλοντική χρήση.

Οι βασικές καταστάσεις για κάθε slice κάθε χρονική στιγμή είναι:

- **activated slice**: Το slice είναι ενεργοποιημένο την παρούσα χρονική στιγμή από τον ερευνητή.

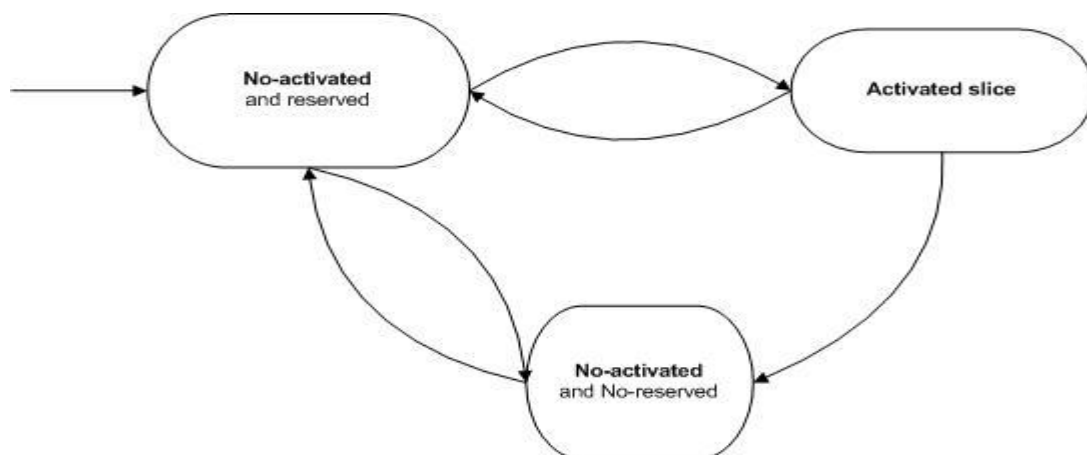
Κεφάλαιο 5: Περιγραφή Υπηρεσιών Ιστού

- **No-activated slice:** Το slice δεν είναι ενεργοποιημένο την παρούσα χρονική στιγμή από τον ερευνητή.

Η κατάσταση No-activated χωρίζεται σε δύο επιμέρους καταστάσεις έτσι ώστε να γίνεται γνωστό αν το slice είναι δεσμευμένο (reserved) για ένα ή περισσότερα χρονικά διαστήματα στο μέλλον ή αν δεν έχει δεσμευθεί (no-reserved) για κάποιο χρονικό διάστημα στο μέλλον. Αυτές οι δύο καταστάσεις είναι αντίστοιχα οι:

1. **No-activated and reserved**
2. **No-activated and No-reserved**

Ας περιγράψουμε τι συμβαίνει με τα διάφορα slices. Όταν ο ερευνητής φτιάχνει το slice του και το δεσμεύει για κάποιο χρονικό διάστημα τότε αυτό βρίσκεται στην κατάσταση **No-activated and reserved**. Στη συνέχεια το slice μπορεί να περάσει σε δυο καταστάσεις. Αν ακυρωθούν οι κρατήσεις που έχουν γίνει για το συγκεκριμένο slice πριν ενεργοποιηθεί τότε αυτό περνάει στην κατάσταση **No-activated and No-reserved**. Ένα slice που ανήκει σε αυτή την κατάσταση μπορεί να δεσμευθεί για κάποιο χρονικό διάστημα από τον ερευνητή και να περάσει στην κατάσταση **No-activated and reserved** και στην συνέχεια στην κατάσταση **activated**. Αν το slice ενεργοποιηθεί (δεν ακυρωθούν τα reservations) τότε περνάει σε κατάσταση **activated**. Μετά το τέλος του χρονικού διαστήματος ενεργοποίησης μπορεί να περάσει σε δύο καταστάσεις. Αν έχει ήδη δεσμευθεί και για ένα ή περισσότερα χρονικά διαστήματα στο μέλλον τότε περνάει στην κατάσταση **No-activated and reserved** και περιμένει να γίνει **activated** την κατάλληλη χρονική στιγμή. Αν το **activated** slice μετά το τέλος του χρονικού διαστήματος ενεργοποίησης δεν έχει δεσμευθεί για μελλοντική χρήση τότε περνάει στην κατάσταση **No-activated and No-reserved**, την οποία περιγράψαμε προηγουμένως. Η ακολουθία των καταστάσεων φαίνεται στο παρακάτω διάγραμμα:



Σχήμα 5.1: Διάγραμμα καταστάσεων των slices

5.1.2 Τύποι δεδομένων

Επιπλέον πρέπει να αναλύσουμε τους τύπους δεδομένων που ανταλλάσσονται μεταξύ πελάτη εξυπηρετητή κατά τη λειτουργία των υπηρεσιών διαδικτύου. Εκτός από τους απλούς τύπους δεδομένων, όπως integer ή strings, έχουμε εισάγει τρεις σύνθετους τύπους δεδομένων, οι οποίοι περιγράφονται στα αντίστοιχα wsdl αρχεία που περιγράφουν τις υπηρεσίες διαδικτύου. Αυτοί οι τρεις τύποι είναι οι εξής:

- **RSpec**

Ο σύνθετος αυτός τύπος (complex type) έχει την παρακάτω δομή και χρησιμοποιείται κατά την δημιουργία ή την τροποποίηση ενός συγκεκριμένου slice:

```
class RSpec {  
    public $name;  
    public $owner;  
    public $timestamp;  
    public $duration;  
    public $rspec1;  
}
```

Όπως παρατηρούμε ο παραπάνω σύνθετος τύπος περιέχει τα στοιχεία \$name,\$owner,\$timestamp,\$duration και \$rspec1. Το στοιχείο \$name είναι το όνομα του slice που θέλουμε να δημιουργήσουμε ή να τροποποιήσουμε. Το στοιχείο \$owner είναι το username του ερευνητή ο οποίος δημιούργησε το συγκεκριμένο slice. Το στοιχείο \$timestamp είναι η χρονική στιγμή έναρξης της χρησιμοποίησης του slice από τον ερευνητή. Το στοιχείο \$duration είναι η χρονική διάρκεια σε ώρες που ο χρήστης επιθυμεί να χρησιμοποιήσει το συγκεκριμένο slice. Τα τελευταία δύο στοιχεία που αναφέραμε χρησιμοποιούνται μόνο στην περίπτωση της δημιουργίας ενός slice (mkslice) από το χρήστη που είναι μια διαδικασία εξαρτώμενη από το χρόνο, αφού όπως θα δούμε και παρακάτω ο χρήστης επιλέγει την χρονική στιγμή που θέλει να χρησιμοποιήσει το slice και τη χρονική διάρκεια, και επιλέγει τους πόρους που είναι διαθέσιμοι εκείνο το χρονικό διάστημα για να προσθέσει στο slice του. Τέλος το πιο σημαντικό στοιχείο είναι το \$rspec1 που αποτελεί ένα XML string το οποίο είναι το RSpec Request που στέλνει ο χρήστης για τους πόρους του συγκεκριμένου slice. Το συγκεκριμένο RSpec συμμορφώνεται τόσο με το σχήμα Core RSpec v.2 Request όσο και με το σχήμα επέκτασης που δημιουργήσαμε για την περιγραφή ασύρματων πόρων. Περιέχει τους πόρους και τα χαρακτηριστικά που ο χρήστης θέλει να χρησιμοποιήσει στο συγκεκριμένο slice του. Οι πληροφορίες του XML string επεξεργάζονται όπως θα δούμε από την υπηρεσία και δημιουργείται ή τροποποιείται το συγκεκριμένο slice.

- **slice**

Ο σύνθετος αυτός τύπος έχει την παρακάτω δομή και χρησιμοποιείται για να περιγράψει τα χαρακτηριστικά του συγκεκριμένου slice:

```
class slice {  
    public $name;  
    public $owner;  
    public $time_start;  
    public $time_end;  
    public $duration;  
    public $rspec;  
}
```

Όπως παρατηρούμε ο παραπάνω σύνθετος τύπος περιέχει τα στοιχεία \$name,\$owner,\$time_start,\$time_end,\$duration και \$rspec. Το στοιχείο \$name είναι το όνομα του συγκεκριμένου slice στο οποίο αναφερόμαστε. Το στοιχείο \$owner είναι το username του ερευνητή ο οποίος δημιούργησε και διαχειρίζεται το συγκεκριμένο slice. Το στοιχείο \$time_start περιλαμβάνει τη χρονική στιγμή έναρξης της χρησιμοποίησης του slice από τον ερευνητή. Το στοιχείο \$time_end περιλαμβάνει τη χρονική στιγμή λήξης της χρησιμοποίησης του slice από τον ερευνητή. Το στοιχείο \$duration είναι η χρονική διάρκεια σε ώρες που ο χρήστης χρησιμοποιεί το συγκεκριμένο slice αν αυτό είναι reserved ή activated. Τέλος το πιο σημαντικό στοιχείο είναι το \$rspec που αποτελεί ένα XML string το οποίο είναι τύπου RSpec v.2 Manifest. Αυτό το RSpec διαμορφώνεται από την υπηρεσία διαδικτύου με βάση τις πληροφορίες που περιέχει η βάση δεδομένων inventory2 σχετικά με το συγκεκριμένο slice, ώστε να ενημερώσει για το ποιό πόροι και με ποιά χαρακτηριστικά έχουν εκχωρηθεί στο συγκεκριμένο slice από το χρήστη του. Το συγκεκριμένο RSpec συμορφώνεται τόσο με το σχήμα Core RSpec v.2 Manifest όσο και με το σχήμα επέκτασης που δημιουργήσαμε για την περιγραφή ασύρματων πόρων.

- **testbedslices**

Ο σύνθετος αυτός τύπος έχει την παρακάτω δομή και χρησιμοποιείται για να συμπεριλάβει ένα σύνολο από slices:

```
class testbedslices {  
    public $slicearray;  
    public $validatearray;  
}
```

Όπως παρατηρούμε ο παραπάνω σύνθετος τύπος περιλαμβάνει τα στοιχεία \$slicearray και \$validatearray. Το στοιχείο \$slicearray είναι ένας πίνακας αποτελούμενος από στοιχεία σύνθετου τύπου slice, που περιγράψαμε παραπάνω, και περιέχει όλα τα slices με τα επιμέρους χαρακτηριστικά τους.. Το στοιχείο \$validatearray είναι ένας πίνακας για τον έλεγχο της εγκυρότητας των RSpec v.2

Manifest που περιέχονται σε κάθε αντικείμενο slice. Εφόσον το RSpec του slice είναι έγκυρο βάζουμε στον πίνακα το στοιχείο 1 αλλιώς το στοιχείο 2. Επομένως αν ο πίνακας \$validatearray έχει μόνο στοιχεία 1 τότε όλα τα RSpec τύπου Manifest είναι έγκυρα κάτι που είναι πολύ σημαντικό για να προχωρήσουμε στην περαιτέρω επεξεργασία τους.

5.2 Υπηρεσίες Ιστού (Web Services)

Οι υπηρεσίες ιστού περιέχουν τις κατάλληλες κλήσεις SQL [37] στην βάση δεδομένων inventory2 για την εξαγωγή των απαραίτητων πληροφοριών σχετικά με τα slices και τη διαθεσιμότητα των πόρων. Ας εξετάσουμε αναλυτικότερα τι λειτουργίες (operations) περιλαμβάνουν οι υπηρεσίες αυτές.

5.2.1 loginservice-Ταυτοποίηση Χρηστών

Η υπηρεσία αυτή είναι υπεύθυνη για την ταυτοποίηση των χρηστών που χρησιμοποιούν το γραφικό περιβάλλον. Η υπηρεσία αυτή σχεδιάστηκε με το WSO2 WSF extension και μάλιστα χωρίς χρήση WSDL αρχείου και περιλαμβάνει τη λειτουργία login.

login Operation

Με βάση αυτή τη λειτουργία πιστοποιούνται τα στοιχεία username και password του χρήστη. Συγκεκριμένα η λειτουργία αυτή δέχεται ως παράμετρο ένα WSMMessage από την πλευρά του χρήστη που περιέχει το username και το password. Στη συνέχεια ελέγχει αν ο συνδυασμός αυτός υπάρχει στον πίνακα login της βάσης δεδομένων inventory2 που βρίσκονται όλοι οι πιστοποιημένοι συνδυασμοί username/password για τους χρήστες της πειραματικής πλατφόρμας. Αν υπάρχει τότε επιστρέφει κατάλληλο μήνυμα WSMMessage ώστε να επιτρέψει την είσοδο του στο user interface και τη διαχείριση των δικών του slices. Σε αντίθετη περίπτωση δεν θα τον αφήσει να εισέλθει στο user interface. Το username του χρήστη ταυτίζεται με το πεδίο owner των slices που έχει δημιουργήσει. Έτσι παίρνοντας με php \$_SESSION το username ο χρήστης θα μπορεί να έχει πρόσβαση μόνο στα δικά του slices.

5.2.2 mksliceservice-Δημιουργία/Τροποποίηση slice

Η υπηρεσία αυτή είναι υπεύθυνη για τη δημιουργία ενός slice από τον ερευνητή καθώς και για την τροποποίηση ενός ήδη υπάρχοντος slice του ερευνητή. Η υπηρεσία και οι λειτουργίες της περιγράφονται κατάλληλα από το sliceoperations.wsdl αρχείο. Περιλαμβάνει δύο λειτουργίες τη mkslice για την δημιουργία ενός slice και τη modifyslice για την τροποποίηση ενός slice.

mkslice Operation

Με αυτή τη λειτουργία της υπηρεσίας ο χρήστης μπορεί να δημιουργήσει ένα καινούργιο slice το οποίο θα περιέχει κάποιους από τους διαθέσιμους πόρους της πειραματικής πλατφόρμας στο χρονικό διάστημα που θέλει να το χρησιμοποιήσει. Η

Κεφάλαιο 5: Περιγραφή Υπηρεσιών Ιστού

λειτουργία αυτή παίρνει ως είσοδο ένα αντικείμενο τύπου RSpec που περιγράψαμε στο παραπάνω υποκεφάλαιο, το οποίο περιέχει το όνομα του slice, το χρήστη του, τη χρονική στιγμή έναρξης χρήσης του slice, τη διάρκεια χρήσης καθώς και το RSpec Request που στέλνει ο χρήστης, προκειμένου να ζητήσει συγκεκριμένους πόρους για το slice του και επιστρέφει ένα string για να ενημερώσει το χρήστη αν οι διαδικασίες έγιναν σωστά.

Το RSpec Request είναι σε μορφή XML string και επεξεργάζεται χρησιμοποιώντας την συνάρτηση SimpleXMLElement της php. Με αυτό τον τρόπο ο εξυπηρετητής ενημερώνεται για το ποιούς πόρους επιθυμεί ο χρήστης στο συγκεκριμένο slice και κάνει τις ανάλογες καταχωρήσεις στους πίνακες της βάσης δεδομένων inventory2. Αρχικά δημιουργεί μια καινούργια καταχώρηση στον πίνακα slices για το καινούργιο slice με όνομα αυτό που έδωσε ο χρήστης και περιέχεται στο στοιχείο name του αντικειμένου RSpec και owner το username του χρήστη που περιέχεται στο στοιχείο owner του αντικειμένου RSpec. Στη συνέχεια έχοντας επεξεργαστεί τους πόρους που έχει ζητήσει ο ερευνητής για το slice του κάνει τις απαραίτητες καταγραφές στους πίνακες slice_node_association και node_if_association. Επιπλέον δημιουργεί μια καινούργια καταχώρηση στον πίνακα reservations για το συγκεκριμένο slice για να το δεσμεύσει τη χρονική στιγμή που έχει ορίσει στο αντικείμενο RSpec και για το δεδομένο χρονικό διάστημα που θέλει δημιουργώντας παράλληλα και τα MySQL events που θα ενεργοποιήσουν το slice εκείνη τη χρονική στιγμή και θα το απενεργοποιήσουν μετά το πέρας του διαστήματος του reservation. Αυτά τα δύο MySQL events θα κάνουν το μεν πρώτο το flag του activate 1 στην καταχώρηση του πίνακα slices τη χρονική στιγμή έναρξης για να δηλώσουμε ότι είναι ενεργοποιημένο το συγκεκριμένο slice και χρησιμοποιείται από τον ερευνητή, ενώ το δεύτερο μετά το πέρας του διαστήματος χρησιμοποίησης θα ξανακάνει το flag activate ίσο με 0 για να δηλώσουμε ότι είναι ανενεργό.

modifyslice Operation

Με αυτή τη λειτουργία της υπηρεσίας ο χρήστης μπορεί να τροποποιήσει ένα οποιαδήποτε slice του και να του προσθέσει ή να του αφαιρέσει πόρους. Αξίζει να επισημάνουμε ότι στην περίπτωση που το slice είναι είτε ενεργοποιημένο είτε δεσμευμένο για μελλοντική χρήση ο χρήστης μπορεί να προσθέσει σε αυτό μόνο πόρους, δηλαδή κόμβους ή κανάλια μετάδοσης, που είναι διαθέσιμοι για εκείνο το χρονικό διάστημα που είναι ενεργοποιημένο ή το έχει δεσμεύσει, έτσι ώστε να μην παρενοχλεί τυχόν άλλα πειράματα που διεξάγονται μέσα στο ίδιο χρονικό διάστημα. Η λειτουργία αυτή παίρνει ως είσοδο ένα αντικείμενο τύπου RSpec που θα περιέχει το όνομα του slice καθώς και το RSpec Request που στέλνει ο χρήστης προκειμένου να τροποποιήσει το slice του. Επιστρέφει ως απάντηση ένα string που ενημερώνει το χρήστη αν έγιναν σωστά οι διαδικασίες.

Το RSpec Request είναι σε μορφή XML string και επεξεργάζεται χρησιμοποιώντας την συνάρτηση SimpleXMLElement της php. Με αυτό τον τρόπο ο

εξυπηρετητής ενημερώνεται για το ποιούς πόρους επιθυμεί ο χρήστης να εκχωρήσει στο συγκεκριμένο slice και κάνει τις ανάλογες καταχωρήσεις στους πίνακες slice_node_association και node_if_association της βάσης δεδομένων inventory2.

5.2.3 *reservesliceservice*-Κράτηση ενός slice για μελλοντική χρήση

Η υπηρεσία αυτή είναι υπεύθυνη για την κράτηση ενός slice για μελλοντική χρήση από τον ερευνητή. Η υπηρεσία αυτή σχεδιάστηκε με το WSO2 WSF extension και μάλιστα χωρίς χρήση WSDL αρχείου και περιλαμβάνει τη λειτουργία reserveslice.

reserveslice Operation

Με βάση αυτή τη λειτουργία γίνεται έλεγχος αν μπορεί ο χρήστης να κάνει κράτηση ενός slice που περιέχει κάποιους πόρους για μελλοντική χρήση. Ο έλεγχος έγκειται στο γεγονός ότι μπορεί κάποιιοι από τους πόρους (κόμβοι και κανάλια μετάδοσης) που ανήκουν σε ένα slice να χρησιμοποιούνται από κάποιο άλλο slice που έχει κρατηθεί είτε για εκείνη είτε για κάποια ενδιάμεση χρονική περίοδο. Αυτό το slice μπορεί να ανήκει είτε στον ίδιο χρήστη είτε σε κάποιον άλλον χρήστη. Με τον έλεγχο αυτόν που κάνουμε αποφεύγουμε την ταυτόχρονη χρησιμοποίηση πόρων από κάποιο άλλο slice. Η λειτουργία δέχεται ως παράμετρο ένα WSMMessage που περιέχει το όνομα του slice, τον owner του, τη χρονική στιγμή έναρξης και τη χρονική διάρκεια σε ώρες που θέλουμε να το χρησιμοποιήσουμε. Αφού επεξεργαστούμε τα στοιχεία που έχει αυτό το WSMMessage και με τις καταγραφές που έχουμε στα tables reservations, slices.slice_node_association και node_if_association κάνουμε τρεις ελέγχους παίρνοντας κατάλληλες πληροφορίες με κλήσεις SQL από αυτούς τους πίνακες της βάσης δεδομένων.

Αρχικά γίνεται έλεγχος αν το ίδιο το slice έχει κρατηθεί στο ίδιο ή σε κάποιο ενδιάμεσο χρονικό διάστημα μέσω του πίνακα reservations. Στη συνέχεια γίνεται έλεγχος αν υπάρχει χρονική επικάλυψη χρήσης κάποιων πόρων του slice από κάποιο άλλο slice εκείνο το χρονικό διάστημα. Όπως αναφέραμε και παραπάνω οι πόροι που διαμοιράζονται χρονικά στους χρήστες είναι οι κόμβοι καθώς και το σύνολο των καναλιών μετάδοσης που ο χρήστης επιλέγει να ορίσει στις ασύρματες διεπαφές που περιέχονται στο slice του. Ως εκ τούτου αρχικά ελέγχουμε αν γίνεται χρονική επικάλυψη χρήσης κάποιου κόμβου του slice από άλλο slice και στη συνέχεια αν γίνεται χρονική επικάλυψη κάποιου καναλιού του slice από άλλο slice που έχει δεσμευθεί το συγκεκριμένο ή κάποιο ενδιάμεσο χρονικό διάστημα. Αν περάσουμε επιτυχώς αυτούς τους τρεις ελέγχους τότε ο χρήστης δεσμεύει το slice του για το χρονικό διάστημα που επιθυμεί. Συγκεκριμένα δημιουργείται μια νέα καταχώρηση στον πίνακα reservations για αυτό το slice και δημιουργούνται τα MySQL events που θα ενεργοποιήσουν το slice εκείνη τη χρονική στιγμή και θα το απενεργοποιήσουν μετά το πέρας του διαστήματος του reservation. Αυτά τα δύο MySQL events θα κάνουν το μεν πρώτο το flag του activate 1 στην καταχώρηση του

Κεφάλαιο 5: Περιγραφή Υπηρεσιών Ιστού

πίνακα slices τη χρονική στιγμή έναρξης για να δηλώσουμε ότι είναι ενεργοποιημένο το συγκεκριμένο slice και χρησιμοποιείται από τον ερευνητή, ενώ το δεύτερο μετά το πέρας του διαστήματος χρησιμοποίησης θα ξανακάνει το flag activate ίσο με 0 για να δηλώσουμε ότι είναι ανενεργό. Η υπηρεσία σε αυτή την περίπτωση επιστρέφει μήνυμα σε WSMMessage ότι η διαδικασία κράτησης ολοκληρώθηκε επιτυχώς. Σε περίπτωση που δεν περάσει κάποιον από τους τρεις αυτούς ελέγχους τότε αποστέλλεται μέσα σε WSMMessage με αναλυτικές πληροφορίες σχετικά με το ποια slices χρησιμοποιούν τους αντίστοιχους πόρους εντός του χρονικού διαστήματος και τότε ξεκινά και τελειώνει η χρησιμοποίηση των slices αυτών ώστε να μπορέσει ο χρήστης να επιλέξει κάποια άλλη ώρα αν θέλει το συγκεκριμένο slice. Συνεπώς μέσω αυτής της υπηρεσίας γίνεται και ένα είδος scheduling των πειραμάτων αφού δεν επιτρέπεται η ταυτόχρονη χρήση των πόρων από τα slices των ερευνητών.

5.2.4 *deactivateservice*-Απενεργοποίηση ενός *activated slice*

Η υπηρεσία αυτή είναι υπεύθυνη για την απενεργοποίηση ενός ενεργοποιημένου (*activated*) slice από τον ερευνητή που το έχει ενεργοποιήσει και έχει τη διαχείριση του. Η υπηρεσία αυτή σχεδιάστηκε με το WSO2 WSF extension και μάλιστα χωρίς χρήση WSDL αρχείου και περιλαμβάνει τη λειτουργία *deactivateslice*.

deactivateslice Operation

Με αυτή τη λειτουργία γίνεται η απενεργοποίηση κάποιου ενεργοποιημένου slice του ερευνητή. Η λειτουργία δέχεται ως παράμετρο ένα WSMMessage από την πλευρά του χρήστη που περιέχει το όνομα του slice του χρήστη που αυτός επιθυμεί να απενεργοποιήσει. Η υπηρεσία με βάση το όνομα του slice χρησιμοποιεί κάποιες κλήσεις SQL για να αλλάξει κάποιες καταγραφές στη βάση δεδομένων. Αρχικά αλλάζει το activate flag στην καταγραφή του πίνακα slices από 1 σε 0 για να δηλώσει ότι το slice δεν είναι ενεργοποιημένο πλέον. Στη συνέχεια αλλάζει τη χρονική στιγμή λήξης στην αντίστοιχη καταγραφή του πίνακα reservations που περιλαμβάνει και τη θέτει ίση με τη χρονική στιγμή που απενεργοποιήσαμε το slice ώστε να μπορούν οι ερευνητές να χρησιμοποιήσουν του πόρους του slice εφόσον αυτό απενεργοποιήθηκε. Τέλος διαγράφουμε το MySQL event που είχε δημιουργηθεί για την απενεργοποίηση του slice. Η λειτουργία επιστρέφει στον χρήστη ένα μήνυμα μέσα σε WSMMessage για επιτυχή εκτέλεση όλων των παραπάνω διαδικασιών.

5.2.5 *cancelreservationservice*-Ακύρωση της κράτησης ενός *slice*

Η υπηρεσία αυτή είναι υπεύθυνη για την ακύρωση της κράτησης (*reservation*) ενός συγκεκριμένου slice από τον ερευνητή για μελλοντική χρήση. Η υπηρεσία αυτή σχεδιάστηκε με το WSO2 WSF extension και μάλιστα χωρίς χρήση WSDL αρχείου και περιλαμβάνει τη λειτουργία *cancelreservations*.

cancelreservations Operation

Με αυτή τη λειτουργία πραγματοποιείται η ακύρωση της κράτησης που έχει γίνει σε ένα συγκεκριμένο slice για μια χρονική περίοδο στο μέλλον. Συγκεκριμένα η υπηρεσία δέχεται ως παράμετρο ένα WSMMessage που περιέχει το όνομα του slice και τη χρονική στιγμή έναρξης της συγκεκριμένης κράτησης. Αφού επεξεργαστεί τις πληροφορίες αυτές από το WSMMessage εκτελεί τις κατάλληλες κλήσεις SQL στη βάση ώστε να ακυρώσει την κράτηση από τον πίνακα reservations. Αρχικά διαγράφει την καταχώριση του πίνακα reservations για το συγκεκριμένο slice και τη συγκεκριμένη χρονική στιγμή έναρξης ενεργοποίησης και στη συνέχεια διαγράφει και τα αντίστοιχα MySQL events που έχουν δημιουργηθεί για να ενεργοποιήσουν το slice εκείνη τη χρονική στιγμή και να το απενεργοποιήσουν μετά το πέρας του διαστήματος του reservation. Η λειτουργία επιστρέφει στον χρήστη ένα μήνυμα μέσα σε WSMMessage για επιτυχή εκτέλεση όλων των παραπάνω διαδικασιών.

5.2.6 delslice-service- Διαγραφή ενός slice

Η υπηρεσία αυτή είναι υπεύθυνη για την διαγραφή ενός οποιουδήποτε slice που έχει δημιουργήσει ο ερευνητής. Η υπηρεσία αυτή σχεδιάστηκε με το WSO2 WSF extension και μάλιστα χωρίς χρήση WSDL αρχείου και περιλαμβάνει τη λειτουργία delslice.

delslice Operation

Με τη λειτουργία αυτή πραγματοποιείται η διαγραφή ενός slice του ερευνητή. Η λειτουργία δέχεται ως παράμετρο ένα WSMMessage που περιέχει το όνομα του συγκεκριμένου slice. Αφού εξαγάγει από το WSMMessage το όνομα του slice εκτελεί τις απαραίτητες κλήσεις SQL (DELETE) για τη διαγραφή από τη βάση δεδομένων όλων των καταγραφών που αφορούν το συγκεκριμένο slice του ερευνητή. Διαγράφει λοιπόν τις καταχωρήσεις από τους πίνακες slices, reservations, slice_node_association, node_if_association που αφορούν το συγκεκριμένο slice καθώς και τα MySQL events που είχαν δημιουργηθεί για μελλοντική ενεργοποίηση και απενεργοποίηση του slice. Η λειτουργία επιστρέφει στον χρήστη ένα μήνυμα μέσα σε WSMMessage για επιτυχή διαγραφή όλων των καταχωρήσεων που αφορούν το συγκεκριμένο slice από τη βάση δεδομένων inventory2.

5.2.7 listslices-service- Λίστα των slices και των διαθέσιμων πόρων

Η υπηρεσία αυτή είναι υπεύθυνη για να επιστρέψει στον χρήστη μια λίστα των slices που ανήκουν σε μια από τις υποκατηγορίες που αναλύσαμε σε προηγούμενο υποκεφάλαιο καθώς και λίστα των πόρων (κόμβοι και κανάλια μετάδοσης) οι οποίοι είναι διαθέσιμοι για κάποιο χρονικό διάστημα ή για κάποιο δεδομένο slice το οποίο είναι reserved ή activated. Η υπηρεσία και οι διάφορες

Κεφάλαιο 5: Περιγραφή Υπηρεσιών Ιστού

λειτουργίες της περιγράφονται κατάλληλα από το listslices12.wsdl αρχείο. Η υπηρεσία περιλαμβάνει 13 λειτουργίες τις οποίες θα αναλύσουμε παρακάτω:

getListSlices Operation

Η λειτουργία αυτή είναι υπεύθυνη για να επιστρέψει στο χρήστη τις πληροφορίες για όλα τα slices που έχουν δημιουργηθεί από όλους τους χρήστες. Δεν δέχεται κάποια παράμετρο εισόδου αλλά επιστρέφει στο χρήστη ένα αντικείμενο τύπου testbedslices, το οποίο περιγράψαμε σε παραπάνω υποκεφάλαιο, και το οποίο περιέχει ένα πίνακα αντικειμένων slice τον slicearray και ένα πίνακα τον validatearray για να δούμε αν τα διάφορα RSpec που περιγράφουν τα στοιχεία των slices είναι έγκυρα. Ο πίνακας slicearray έχει ως στοιχεία κάποια αντικείμενα τύπου slice. Το κάθε αντικείμενο τύπου slice περιέχει διάφορα στοιχεία για το slice, όπως όνομα και χρήστη, τα οποία περιγράψαμε παραπάνω. Τα στοιχεία αυτά τα εξαγάγει η υπηρεσία με κατάλληλες κλήσεις SQL στους πίνακες της βάσης δεδομένων. Επιπλέον κάθε αντικείμενο τύπου slice περιλαμβάνει ένα RSpec v.2 τύπου Manifest στο οποίο περιέχονται όλοι οι πόροι και τα χαρακτηριστικά τους που έχουν εκχωρηθεί στο συγκεκριμένο slice από τον χρήστη του. Το RSpec αυτό διαμορφώνεται στην υπηρεσία με κατάλληλες κλήσεις SQL στους πίνακες της βάσης δεδομένων, που περιέχουν πληροφορίες για τους πόρους που έχουν εκχωρηθεί στο slice. Συνεπώς με αυτή τη λειτουργία επιστρέφεται στο χρήστη μια λίστα με όλα τα slices και τα επιμέρους στοιχεία που έχουν εκχωρηθεί σε κάθε ένα από αυτά, καθώς και ένα validatearray, για να πιστοποιήσουμε στο χρήστη ότι τα διάφορα RSpec v.2 Manifest είναι καλώς δομημένα στην πλευρά τους εξυπηρετητή και επομένως μπορεί να τα χρησιμοποιήσει για να πάρει πληροφορίες για τα slices.

getListMySlices Operation

Η λειτουργία αυτή κάνει ότι ακριβώς και η προηγούμενη λειτουργία getListSlices με τη διαφορά ότι επιστρέφει μόνο τα slices που έχει δημιουργήσει ένας συγκεκριμένος χρήστης. Δέχεται ως παράμετρο εισόδου το username του συγκεκριμένου ερευνητή ώστε να επιστρέψει πληροφορίες σχετικά με τα slices που έχει δημιουργήσει ο ερευνητής με το συγκεκριμένο username. Στη συνέχεια εκτελεί την ίδια διαδικασία που περιγράψαμε και στην παραπάνω λειτουργία και δημιουργεί τον πίνακα slicearray με στοιχεία slice αντικείμενα, το καθένα από τα οποία περιγράφει ένα slice του συγκεκριμένου ερευνητή. Επιπλέον δημιουργεί και το validatearray για τον έλεγχο της εγκυρότητας των διάφορων RSpec v.2 Manifest. Τέλος στέλνει στον χρήστη το αντικείμενο τύπου testbedslices το οποίο περιέχει τις απαραίτητες πληροφορίες για τα διάφορα slices που έχει δημιουργήσει.

getListReservedSlices Operation

Η λειτουργία αυτή είναι υπεύθυνη για να επιστρέψει στον χρήστη πληροφορίες για όλα τα slices τα οποία είναι reserved ανεξαρτήτως σε ποιόν ερευνητή ανήκουν. Reserved, όπως αναφέραμε και παραπάνω, είναι τα slices τα οποία έχουν δεσμευθεί για μελλοντική χρήση. Προκειμένου να βρούμε ποια από τα slices του testbed είναι reserved η λειτουργία εκτελεί τις κατάλληλες κλήσεις SQL

Κεφάλαιο 5: Περιγραφή Υπηρεσιών Ιστού

στον πίνακα reservations. Η συγκεκριμένη λειτουργία δεν δέχεται κάποια παράμετρο εισόδου και επιστρέφει στο χρήστη ένα αντικείμενο τύπου testbedslices, το οποίο περιέχει ένα πίνακα slicearray και ένα πίνακα validatearray. Στον πίνακα slicearray περιέχονται τα αντικείμενα τύπου slice που περιέχουν πληροφορίες για τα reserved slices των ερευνητών. Κάθε αντικείμενο slice θα περιέχει το όνομα του slice στο στοιχείο \$name, τον χρήστη του slice στο στοιχείο \$owner, τη χρονική στιγμή έναρξης της κράτησης του στο στοιχείο \$time_start, τη χρονική στιγμή λήξης της κράτησης του στο στοιχείο \$time_end. Αυτές τις πληροφορίες τις εξαγάγει η λειτουργία με κατάλληλες κλήσεις SQL στις καταγραφές των πινάκων slices και reservations που αφορούν το συγκεκριμένο slice. Επιπλέον κάθε αντικείμενο slice θα περιέχει ένα RSpec v.2 Manifest το οποίο θα περιέχει τους πόρους και τα χαρακτηριστικά τους που έχουν εκχωρηθεί στο συγκεκριμένο reserved slice. Αυτά τα εξαγάγει η λειτουργία με κατάλληλες κλήσεις SQL στους πίνακες της βάσης δεδομένων που περιέχουν πληροφορίες για τους πόρους που έχουν εκχωρηθεί στο slice. Συνεπώς με αυτή τη λειτουργία επιστρέφεται στο χρήστη μια λίστα με όλα τα reserved slices και τα επιμέρους στοιχεία που έχουν εκχωρηθεί σε κάθε ένα από αυτά, καθώς και ένα validatearray, για να πιστοποιήσουμε στο χρήστη ότι τα διάφορα RSpec v.2 Manifest είναι καλώς δομημένα στην πλευρά τους εξυπηρετητή και επομένως μπορεί να τα χρησιμοποιήσει για να πάρει πληροφορίες για τα slices.

getListMyReservedSlices Operation

Η λειτουργία αυτή κάνει ότι ακριβώς και η προηγούμενη λειτουργία getListReservedSlices με τη διαφορά ότι επιστρέφει μόνο τα reserved slices που έχει δημιουργήσει ένας συγκεκριμένος χρήστης. Δέχεται ως παράμετρο εισόδου το username του συγκεκριμένου ερευνητή ώστε να επιστρέψει πληροφορίες σχετικά με τα slices που έχει δημιουργήσει ο ερευνητής με το συγκεκριμένο username. Στη συνέχεια εκτελεί την ίδια διαδικασία που περιγράψαμε και στην παραπάνω λειτουργία και δημιουργεί τον πίνακα slicearray με στοιχεία slice αντικείμενα, το καθένα από τα οποία περιγράφει ένα reserved slice του συγκεκριμένου ερευνητή. Επιπλέον δημιουργεί και το validatearray για τον έλεγχο της εγκυρότητας των διάφορων RSpec v.2 Manifest. Τέλος στέλνει στον χρήστη το αντικείμενο τύπου testbedslices το οποίο περιέχει τις απαραίτητες πληροφορίες για τα διάφορα reserved slices που του ανήκουν.

getListActivatedSlices Operation

Η λειτουργία αυτή είναι υπεύθυνη για να επιστρέψει στον χρήστη πληροφορίες για όλα τα slices τα οποία είναι στην κατάσταση activated τη συγκεκριμένη χρονική στιγμή ανεξαρτήτως σε ποιόν ερευνητή ανήκουν. Στην κατάσταση activated, όπως αναφέραμε και παραπάνω, ανήκουν τα slices τα οποία είναι ενεργοποιημένα αυτή τη χρονική στιγμή. Προκειμένου να βρούμε ποιιά slices είναι activated αυτή τη χρονική στιγμή εκτελούμε τις κατάλληλες κλήσεις SQL στον πίνακα slices ώστε να δούμε ποιιά από αυτά έχουν το activate flag ίσο με 1 και επομένως είναι activated. Η συγκεκριμένη λειτουργία δεν δέχεται κάποια παράμετρο εισόδου και επιστρέφει στο χρήστη ένα αντικείμενο τύπου testbedslices, το οποίο

Κεφάλαιο 5: Περιγραφή Υπηρεσιών Ιστού

περιέχει ένα πίνακα `slicearray` και ένα πίνακα `validatearray`. Στον πίνακα `slicearray` περιέχονται τα αντικείμενα τύπου `slice` που περιέχουν πληροφορίες για τα `reserved slices` των ερευνητών. Κάθε αντικείμενο `slice` θα περιέχει το όνομα του `slice` στο στοιχείο `$name`, τον χρήστη του `slice` στο στοιχείο `$owner`, τη χρονική στιγμή ενεργοποίησης του στο στοιχείο `$time_start`, τη χρονική διάρκεια ενεργοποίησης στο στοιχείο `$duration`. Αυτές τις πληροφορίες τις εξαγάγει η λειτουργία με κατάλληλες κλήσεις SQL στην καταχώρηση του πίνακα `slices` που αφορά συγκεκριμένο `slice`. Επιπλέον κάθε αντικείμενο `slice` θα περιέχει ένα `RSpec v.2 Manifest` το οποίο θα περιέχει τους πόρους και τα χαρακτηριστικά τους που έχουν εκχωρηθεί στο συγκεκριμένο `activated slice`. Αυτά τα εξαγάγει η λειτουργία με κατάλληλες κλήσεις SQL στους πίνακες της βάσης δεδομένων που περιέχουν πληροφορίες για τους πόρους που έχουν εκχωρηθεί στο `slice`. Συνεπώς με αυτή τη λειτουργία επιστρέφεται στο χρήστη μια λίστα με όλα τα `activated slices` και τα επιμέρους στοιχεία που έχουν εκχωρηθεί σε κάθε ένα από αυτά, καθώς και ένα `validatearray`, για να πιστοποιήσουμε στο χρήστη ότι τα διάφορα `RSpec v.2 Manifest` είναι καλώς δομημένα στην πλευρά τους εξυπηρετητή και επομένως μπορεί να τα χρησιμοποιήσει για να πάρει πληροφορίες για τα `slices`.

getListMyActivatedSlices Operation

Η λειτουργία αυτή κάνει ότι ακριβώς και η προηγούμενη λειτουργία `getListActivatedSlices` με τη διαφορά ότι επιστρέφει μόνο τα `activated slices` που έχει δημιουργήσει ένας συγκεκριμένος χρήστης. Δέχεται ως παράμετρο εισόδου το `username` του συγκεκριμένου ερευνητή ώστε να επιστρέψει πληροφορίες σχετικά με τα `slices` που έχει δημιουργήσει ο ερευνητής με το συγκεκριμένο `username`. Στη συνέχεια εκτελεί την ίδια διαδικασία που περιγράψαμε και στην παραπάνω λειτουργία και δημιουργεί τον πίνακα `slicearray` με στοιχεία `slice` αντικείμενα, το καθένα από τα οποία περιγράφει ένα `activated slice` του συγκεκριμένου ερευνητή. Επιπλέον δημιουργεί και το `validatearray` για τον έλεγχο της εγκυρότητας των διάφορων `RSpec v.2 Manifest`. Τέλος στέλνει στον χρήστη το αντικείμενο τύπου `testbedslices` το οποίο περιέχει τις απαραίτητες πληροφορίες για τα διάφορα `activated slices` που του ανήκουν.

getListNoReservedSlices Operation

Η λειτουργία αυτή είναι υπεύθυνη για να επιστρέψει στον χρήστη πληροφορίες για όλα τα `slices` τα οποία είναι `no-reserved` ανεξαρτήτως σε ποιόν ερευνητή ανήκουν. `No-reserved`, όπως αναφέραμε και παραπάνω, είναι τα `slices` τα οποία δεν έχουν δεσμευθεί για μελλοντική χρήση. Προκειμένου να βρούμε ποια από τα `slices` του `testbed` είναι `no-reserved` η λειτουργία εκτελεί τις κατάλληλες κλήσεις SQL στον πίνακα `reservations`. Η συγκεκριμένη λειτουργία δεν δέχεται κάποια παράμετρο εισόδου και επιστρέφει στο χρήστη ένα αντικείμενο τύπου `testbedslices`, το οποίο περιέχει ένα πίνακα `slicearray` και ένα πίνακα `validatearray`. Στον πίνακα `slicearray` περιέχονται τα αντικείμενα τύπου `slice` που περιέχουν πληροφορίες για τα `no-reserved slices` των ερευνητών. Κάθε αντικείμενο `slice` θα περιέχει το όνομα του `slice` στο στοιχείο `$name` και τον χρήστη του `slice` στο στοιχείο `$owner`. Αυτές τις

Κεφάλαιο 5: Περιγραφή Υπηρεσιών Ιστού

πληροφορίες τις εξαγάγει η λειτουργία με κατάλληλες κλήσεις SQL στην καταγραφή του πίνακα slices που αφορά το συγκεκριμένο slice. Επιπλέον κάθε αντικείμενο slice θα περιέχει ένα RSpec v.2 Manifest το οποίο θα περιέχει τους πόρους και τα χαρακτηριστικά τους που έχουν εκχωρηθεί στο συγκεκριμένο no-reserved slice. Αυτά τα εξαγάγει η λειτουργία με κατάλληλες κλήσεις SQL στους πίνακες της βάσης δεδομένων που περιέχουν πληροφορίες για τους πόρους που έχουν εκχωρηθεί στο slice. Συνεπώς με αυτή τη λειτουργία επιστρέφεται στο χρήστη μια λίστα με όλα τα reserved slices και τα επιμέρους στοιχεία που έχουν εκχωρηθεί σε κάθε ένα από αυτά, καθώς και ένα validatearray, για να πιστοποιήσουμε στο χρήστη ότι τα διάφορα RSpec v.2 Manifest είναι καλώς δομημένα στην πλευρά τους εξυπηρετητή και επομένως μπορεί να τα χρησιμοποιήσει για να πάρει πληροφορίες για τα slices.

getListMyNoReservedSlices Operation

Η λειτουργία αυτή κάνει ότι ακριβώς και η προηγούμενη λειτουργία getListNoReservedSlices με τη διαφορά ότι επιστρέφει μόνο τα no-reserved slices που έχει δημιουργήσει ένας συγκεκριμένος χρήστης. Δέχεται ως παράμετρο εισόδου το username του συγκεκριμένου ερευνητή ώστε να επιστρέψει πληροφορίες σχετικά με τα slices που έχει δημιουργήσει ο ερευνητής με το συγκεκριμένο username. Στη συνέχεια εκτελεί την ίδια διαδικασία που περιγράψαμε και στην παραπάνω λειτουργία και δημιουργεί τον πίνακα slicearray με στοιχεία slice αντικείμενα, το καθένα από τα οποία περιγράφει ένα no-reserved slice του συγκεκριμένου ερευνητή. Επιπλέον δημιουργεί και το validatearray για τον έλεγχο της εγκυρότητας των διάφορων RSpec v.2 Manifest. Τέλος στέλνει στον χρήστη το αντικείμενο τύπου testbedslices το οποίο περιέχει τις απαραίτητες πληροφορίες για τα διάφορα no-reserved slices που του ανήκουν.

getListNoActivatedSlices Operation

Η λειτουργία αυτή είναι υπεύθυνη για να επιστρέψει στον χρήστη πληροφορίες για όλα τα slices τα οποία ανήκουν στην γενικότερη κατάσταση no-activated τη συγκεκριμένη χρονική στιγμή ανεξαρτήτως σε ποιόν ερευνητή ανήκουν. No-activated, όπως αναφέραμε και παραπάνω, είναι τα slices τα οποία δεν είναι ενεργοποιημένα αυτή τη χρονική στιγμή. Προκειμένου να βρούμε ποιά slices είναι no-activated αυτή τη χρονική στιγμή εκτελούμε τις κατάλληλες κλήσεις SQL στον πίνακα slices ώστε να δούμε ποιά από αυτά έχουν το activate flag ίσο με 0 και επομένως είναι no-activated. Η συγκεκριμένη λειτουργία δεν δέχεται κάποια παράμετρο εισόδου και επιστρέφει στο χρήστη ένα αντικείμενο τύπου testbedslices, το οποίο περιέχει ένα πίνακα slicearray και ένα πίνακα validatearray. Στον πίνακα slicearray περιέχονται τα αντικείμενα τύπου slice που περιέχουν πληροφορίες για τα reserved slices των ερευνητών. Κάθε αντικείμενο slice θα περιέχει το όνομα του slice στο στοιχείο \$name και τον χρήστη του slice στο στοιχείο \$owner. Αυτές τις πληροφορίες τις εξαγάγει η λειτουργία με κατάλληλες κλήσεις SQL στην καταχώρηση του πίνακα slices που αφορά συγκεκριμένο slice. Επιπλέον κάθε αντικείμενο slice θα περιέχει ένα RSpec v.2 Manifest το οποίο θα περιέχει τους πόρους και τα χαρακτηριστικά τους που έχουν εκχωρηθεί στο συγκεκριμένο no-

Κεφάλαιο 5: Περιγραφή Υπηρεσιών Ιστού

activated slice. Αυτά τα εξαγάγει η λειτουργία με κατάλληλες κλήσεις SQL στους πίνακες της βάσης δεδομένων που περιέχουν πληροφορίες για τους πόρους που έχουν εκχωρηθεί στο slice. Συνεπώς με αυτή τη λειτουργία επιστρέφεται στο χρήστη μια λίστα με όλα τα no-activated slices και τα επιμέρους στοιχεία που έχουν εκχωρηθεί σε κάθε ένα από αυτά, καθώς και ένα validatearray, για να πιστοποιήσουμε στο χρήστη ότι τα διάφορα RSpec v.2 Manifest είναι καλώς δομημένα στην πλευρά τους εξυπηρετητή και επομένως μπορεί να τα χρησιμοποιήσει για να πάρει πληροφορίες για τα slices.

getListMyNoActivatedSlices Operation

Η λειτουργία αυτή κάνει ότι ακριβώς και η προηγούμενη λειτουργία getListNoActivatedSlices με τη διαφορά ότι επιστρέφει μόνο τα no-activated slices που έχει δημιουργήσει ένας συγκεκριμένος χρήστης. Δέχεται ως παράμετρο εισόδου το username του συγκεκριμένου ερευνητή ώστε να επιστρέψει πληροφορίες σχετικά με τα slices που έχει δημιουργήσει ο ερευνητής με το συγκεκριμένο username. Στη συνέχεια εκτελεί την ίδια διαδικασία που περιγράψαμε και στην παραπάνω λειτουργία και δημιουργεί τον πίνακα slicearray με στοιχεία slice αντικείμενα, το καθένα από τα οποία περιγράφει ένα no-activated slice του συγκεκριμένου ερευνητή. Επιπλέον δημιουργεί και το validatearray για τον έλεγχο της εγκυρότητας των διάφορων RSpec v.2 Manifest. Τέλος στέλνει στον χρήστη το αντικείμενο τύπου testbedslices το οποίο περιέχει τις απαραίτητες πληροφορίες για τα διάφορα no-activated slices που του ανήκουν.

getAvailableResources Operation

Η λειτουργία αυτή είναι υπεύθυνη για να επιστρέψει στο χρήστη ένα XML string, το οποίο αποτελεί RSpec v.2 τύπου Advertisement ώστε να τον ενημερώσει σχετικά με το ποιοί κόμβοι είναι διαθέσιμοι για την χρονική περίοδο που επιθυμεί να χρησιμοποιήσει το slice που θα δημιουργήσει καθώς ποιά κανάλια μετάδοσης είναι διαθέσιμα για τα interfaces των διαθέσιμων κόμβων. Επιπλέον ενημερώνει το χρήστη και σχετικά με το ποιοί ρυθμοί μετάδοσης υποστηρίζονται από τις ασύρματες διεπαφές καθώς και όλους τους δυνατούς τρόπους λειτουργίας των ασύρματων διεπαφών (AP ή STA). Δέχεται ως παράμετρο τη χρονική στιγμή έναρξης και τη χρονική διάρκεια χρησιμοποίησης του slice. Χρησιμοποιώντας κατάλληλες κλήσεις SQL στους πίνακες της βάσης δεδομένων η υπηρεσία βρίσκει ποιοί κόμβοι και ποιά κανάλια είναι διαθέσιμα εκείνο το χρονικό διάστημα και δεν χρησιμοποιούνται από κάποιο slice που έχει δεσμευθεί εντός αυτού του χρονικού διαστήματος (λειτουργία scheduling).

Παίρνοντας τις πληροφορίες για τους διαθέσιμους πόρους η λειτουργία αυτή διαμορφώνει κατάλληλα ένα RSpec v.2 Advertisement το οποίο θα στείλει μέσω του service στον ερευνητή. Το RSpec v.2 Advertisement θα πρέπει να συμμορφώνεται τόσο με το Core RSpec σχήμα όσο και με το σχήμα επέκτασης αφού για κάθε διεπαφή θα περιέχει και τα διαθέσιμα κανάλια μετάδοσης (RadioChannel), καθώς και τους ρυθμούς μετάδοσης και τους τρόπους λειτουργίας που υποστηρίζει

Κεφάλαιο 5: Περιγραφή Υπηρεσιών Ιστού

κάθε διεπαφή. Ο χρήστης στη συνέχεια καλείται να επιλέξει ένα ρυθμό μετάδοσης, ένα διαθέσιμο κανάλι και ένα τρόπο λειτουργίας για κάθε διεπαφή που θα επιλέξει. Το RSpec v.2 advertisement που διαμορφώνεται είναι ανάλογο με αυτό που παρουσιάσαμε στο κεφάλαιο 3.2.2 και θα περιλαμβάνει τους διαθέσιμους κόμβους με τα αντίστοιχα URNs που τους περιγράφουν, τις διαθέσιμες διεπαφές των κόμβων με τα αντίστοιχα URNs που τα περιγράφουν και τα διαθέσιμα ασύρματα χαρακτηριστικά (ρυθμός μετάδοσης, κανάλι μετάδοσης, τρόπος λειτουργίας) για κάθε διεπαφή. Συνεπώς η λειτουργία αυτή επιστρέφει στο χρήστη το κατάλληλο RSpec v.2 Advertisement έτσι ώστε να ενημερωθεί για τους πόρους που είναι διαθέσιμοι και να δημιουργήσει το slice του.

getListSliceResources Operation

Η λειτουργία αυτή είναι υπεύθυνη για να επιστρέψει στο χρήστη τους πόρους που έχουν εκχωρηθεί σε ένα συγκεκριμένο slice. Δέχεται ως παράμετρο το όνομα του slice και εκτελώντας κατάλληλες κλήσεις SQL στους πίνακες της βάσης δεδομένων inventory2 παίρνει τους αντίστοιχους πόρους που έχουν εκχωρηθεί σε αυτό το slice. Στη συνέχεια διαμορφώνει ένα RSpec v.2 Manifest το οποίο περιέχει τους κόμβους, τις διεπαφές καθώς και τα χαρακτηριστικά των διεπαφών (Rate, RadioChannel, OperatingMode), τα οποία έχουν εκχωρηθεί στο συγκεκριμένο slice. Το RSpec v.2 Manifest που διαμορφώνεται πρέπει να είναι έγκυρο τόσο με βάση το Core RSpec σχήμα όσο και με το σχήμα επέκτασης για τα χαρακτηριστικά των ασύρματων διεπαφών. Η λειτουργία αφού ελέγξει την εγκυρότητα του RSpec το αποστέλλει στο χρήστη έτσι ώστε να ενημερωθεί για τους πόρους που ανήκουν σε ένα συγκεκριμένο slice.

getListAvailableResources Operation

Η λειτουργία αυτή είναι υπεύθυνη για να επιστρέψει στο χρήστη ένα XML string, το οποίο αποτελεί RSpec v.2 τύπου Advertisement ώστε να τον ενημερώσει σχετικά με το ποιοί κόμβοι είναι διαθέσιμοι για ένα συγκεκριμένο slice, ποιά κανάλια μετάδοσης είναι διαθέσιμα για τα interfaces των διαθέσιμων κόμβων, καθώς και ποιοί ρυθμοί μετάδοσης και ποιοί τρόποι λειτουργίας υποστηρίζονται από τις ασύρματες διεπαφές. Χρησιμοποιείται όταν θέλουμε να τροποποιήσουμε ένα slice. Δέχεται ως παράμετρο το όνομα του συγκεκριμένου slice.

Σε περίπτωση που το slice αυτό δεν είναι ούτε ενεργοποιημένο ούτε δεσμευμένο για μελλοντική χρήση (No-activated and No-reserved) η υπηρεσία θα επιστρέψει μέσω RSpec v.2 Advertisement όλους τους κόμβους και όλα τα κανάλια μετάδοσης συμπεριλαμβανομένων και αυτών που ανήκουν στο συγκεκριμένο slice και συνεπώς είναι διαθέσιμα για χρησιμοποίηση από το συγκεκριμένο slice. Χρησιμοποιώντας κατάλληλες κλήσεις SQL στους πίνακες της βάσης δεδομένων η λειτουργία βρίσκει ποιοί κόμβοι και ποιά κανάλια υπάρχουν καθώς και ποιοί ρυθμοί μετάδοσης και ποιοί τρόποι λειτουργίας υποστηρίζονται από τις ασύρματες διεπαφές και τα επιστρέφει διαμορφώνοντας ένα RSpec v.2 Advertisement.

Κεφάλαιο 5: Περιγραφή Υπηρεσιών Ιστού

Σε περίπτωση που το slice αυτό είναι είτε ενεργοποιημένο (activated) είτε δεσμευμένο για μελλοντική χρήση (No-activated and reserved) η υπηρεσία θα επιστρέψει μέσω RSpec v.2 Advertisement τους κόμβους και τα κανάλια μετάδοσης που είναι διαθέσιμα για εκείνο το χρονικό διάστημα που έχει οριστεί η χρησιμοποίηση του συμπεριλαμβανομένων και αυτών που ανήκουν στο συγκεκριμένο slice και συνεπώς είναι διαθέσιμα για χρησιμοποίηση από το συγκεκριμένο slice (λειτουργία scheduling). Χρησιμοποιώντας κατάλληλες κλήσεις SQL στους πίνακες της βάσης δεδομένων η λειτουργία βρίσκει ποιό κόμβοι και ποιά κανάλια είναι διαθέσιμα εκείνο το χρονικό διάστημα και δεν χρησιμοποιούνται από κάποιο slice που έχει δεσμευθεί εντός αυτού του χρονικού διαστήματος.

Παίρνοντας τις πληροφορίες για τους διαθέσιμους πόρους η λειτουργία αυτή διαμορφώνει κατάλληλα ένα RSpec v.2 Advertisement το οποίο θα στείλει μέσω του service στον ερευνητή. Το RSpec v.2 Advertisement θα πρέπει να συμμορφώνεται τόσο με το Core RSpec σχήμα όσο και με το σχήμα επέκτασης αφού για κάθε διεπαφή θα περιέχει και τα διαθέσιμα κανάλια μετάδοσης (RadioChannel), καθώς και τους ρυθμούς μετάδοσης και τους τρόπους λειτουργίας που υποστηρίζει κάθε διεπαφή. Ο χρήστης στη συνέχεια καλείται να επιλέξει ένα ρυθμό μετάδοσης, ένα διαθέσιμο κανάλι και ένα τρόπο λειτουργίας για κάθε διεπαφή που θα επιλέξει. Το RSpec v.2 advertisement που διαμορφώνεται είναι ανάλογο με αυτό που παρουσιάσαμε στο κεφάλαιο 3.2.2. Συνεπώς η λειτουργία αυτή επιστρέφει στο χρήστη το κατάλληλο RSpec v.2 Advertisement έτσι ώστε να ενημερωθεί για τους πόρους που είναι διαθέσιμοι και να τροποποιήσει ένα δεδομένο slice του.

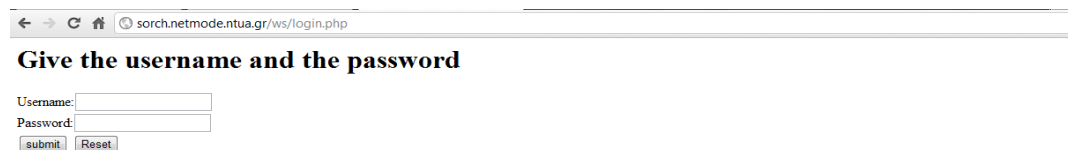
Κεφάλαιο 6

Περιγραφή Γραφικού Περιβάλλοντος Χρήστη (Graphical User Interface)

Προκειμένου να μπορούν οι διάφοροι χρήστες να χρησιμοποιήσουν τις υπηρεσίες που περιγράψαμε στο Κεφάλαιο 5 και να μπορούν να δημιουργήσουν αλλά και να διαχειριστούν τα slices τους δημιουργήσαμε ένα γραφικό περιβάλλον χρήστη. Αυτό περιλαμβάνει διάφορες λειτουργίες που μπορεί να κάνει ένας χρήστης, όπως να δημιουργήσει ένα slice με κάποιους από τους διαθέσιμους πόρους το χρονικό διάστημα που θέλει να το χρησιμοποιήσει, να τροποποιήσει ένα slice ,να δεσμεύσει ένα slice του για μελλοντική χρήση, να απενεργοποιήσει ένα slice,να ακυρώσει κάποιες από τις κρατήσεις του, να διαγράψει ένα slice κτλ χρησιμοποιώντας τις υπηρεσίες διαδικτύου που σχεδιάσαμε.

Για να υλοποιηθεί το user interface χρησιμοποιήσαμε τη γλώσσα HTML και συγκεκριμένα HTML forms υλοποιημένες με τη μέθοδο POST [38]. Αυτές τις σελίδες τις ενσωματώσαμε σε κάποια php scripts τα οποία καλούνε τα απαραίτητα operations των services ώστε να εκτελεστούν οι ενέργειες που θέλουμε.

Αρχικά έχουμε δημιουργήσει ένα php script, το οποίο είναι υπεύθυνο για την ταυτοποίηση των χρηστών και σε αυτό έχουμε ενσωματώσει μια HTML φόρμα:



← → 🏠 🌐 soch.netmode.ntua.gr/ws/login.php

Give the username and the password

Username:

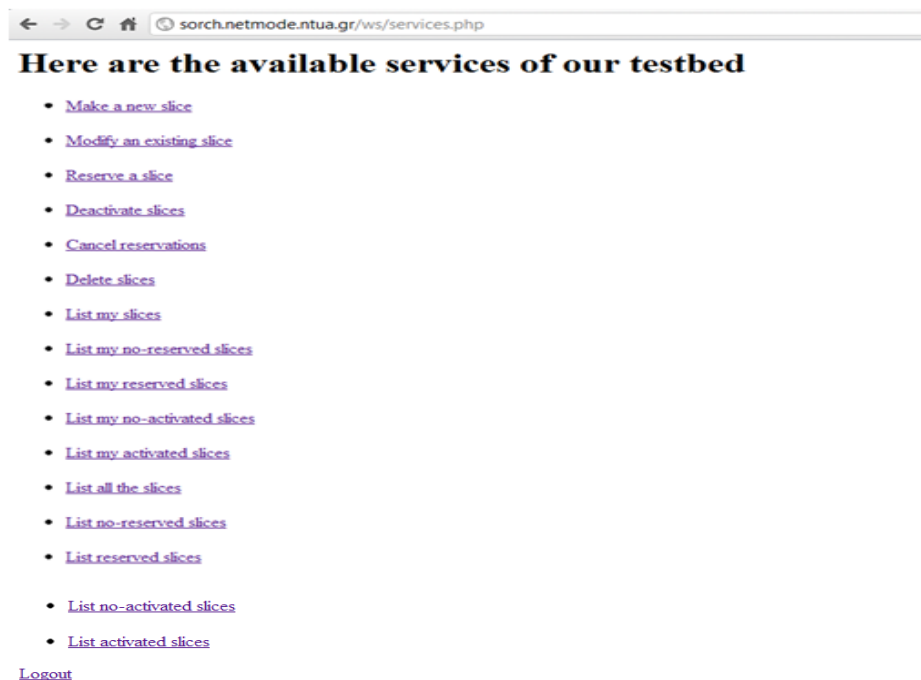
Password:

Σχήμα 6.1: Σελίδα για ταυτοποίηση των χρηστών

Στην παραπάνω σελίδα ο χρήστης πληκτρολογεί το username και το password. Αν ο συνδυασμός είναι σωστός σύμφωνα με την υπηρεσία loginservice που περιγράψαμε στο προηγούμενο κεφάλαιο τότε ο χρήστης με αυτό το username είναι πιστοποιημένος χρήστης της πειραματικής πλατφόρμας και έχει πρόσβαση στο php script στο οποίο έχουμε ενσωματώσει την HTML σελίδα που περιλαμβάνει της διάφορες υπηρεσίες για την δημιουργία και τη διαχείριση των slices. Το username περνάει στα επόμενα php scripts με την php παράμετρο \$_SESSION ώστε να μπορεί ο χρήστης να διαχειριστεί μόνο δικά του slices και να μην επηρεάζει slices των άλλων χρηστών που χρησιμοποιούν την πειραματική πλατφόρμα.

Κεφάλαιο 6: Περιγραφή Γραφικού Περιβάλλοντος Χρήστη (Graphical User Interface)

Το php script που δίνει πρόσβαση στις διαθέσιμες υπηρεσίες διαδικτύου φαίνεται παρακάτω:



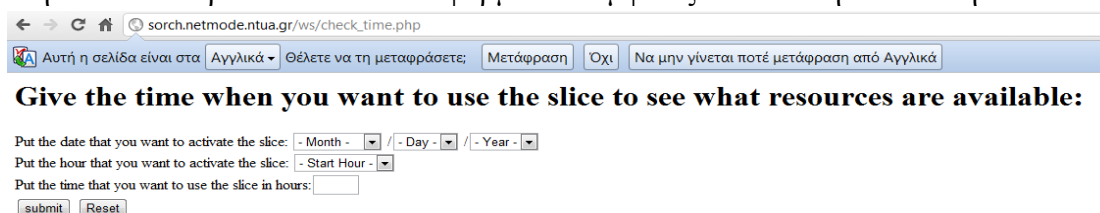
Σχήμα 6.2: Οι διάφορες υπηρεσίες που είναι διαθέσιμες από το γραφικό περιβάλλον

Όπως παρατηρούμε από την παραπάνω εικόνα ο χρήστης έχει πρόσβαση σε μια σειρά από υπηρεσίες. Παρακάτω θα αναλύσουμε πως ακριβώς λειτουργεί η κάθε μία από τις παραπάνω υπηρεσίες.

Make a new slice-Δημιουργία καινούργιου slice

Η λειτουργία αυτή αποτελείται από συνολικά τρία php scripts και χρησιμοποιείται για τη δημιουργία ενός slice από το χρήστη με πόρους που είναι διαθέσιμοι για κάποιο συγκεκριμένο χρονικό διάστημα.

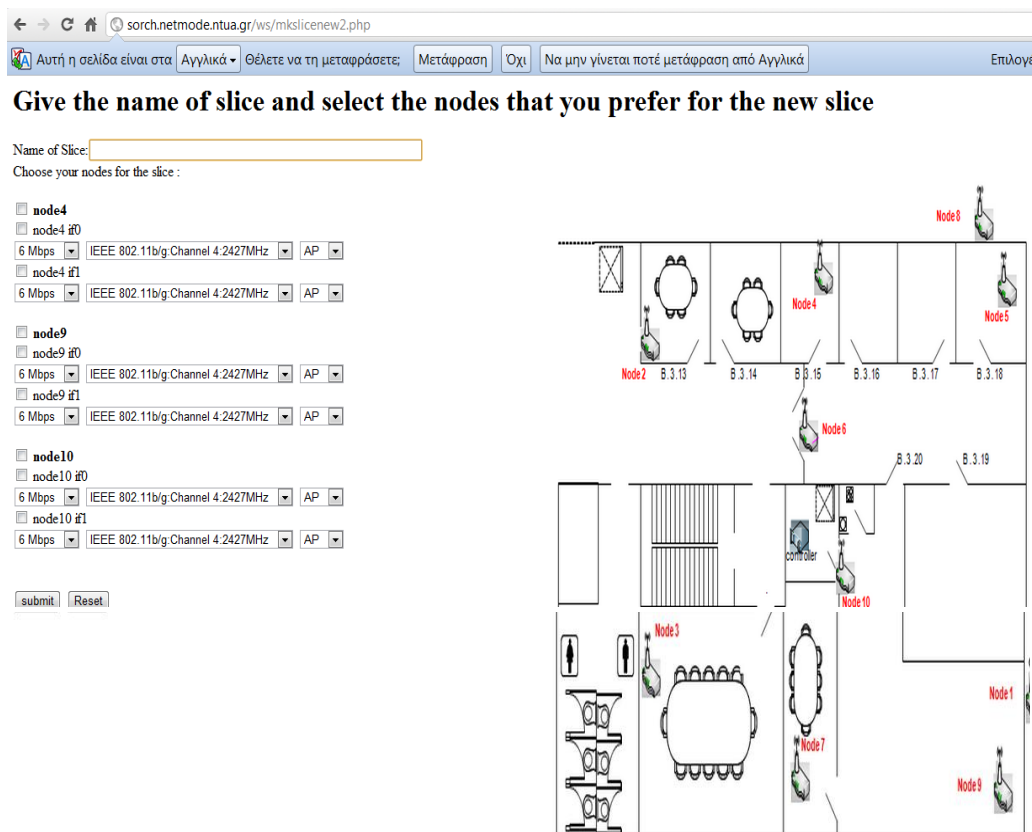
Στο πρώτο php script ενσωματώνεται μια HTML φόρμα και ο χρήστης καλείται να επιλέξει την ημερομηνία, την ώρα έναρξης και τη χρονική διάρκεια σε ώρες που επιθυμεί να χρησιμοποιήσει το slice που θα δημιουργήσει. Απαραίτητη προϋπόθεση είναι η ημερομηνία και η χρονική στιγμή που επιλέγει ο χρήστης να μην ανήκουν στο παρελθόν. Η HTML φόρμα που εμφανίζεται είναι η ακόλουθη:



Σχήμα 6.3: Επιλογή χρονικού διαστήματος χρήσης του slice

Κεφάλαιο 6: Περιγραφή Γραφικού Περιβάλλοντος Χρήστη (Graphical User Interface)

Η HTML φόρμα καλεί ως ACTION το δεύτερο php script στο οποίο θα ενσωματώσουμε άλλη μια HTML φόρμα. Η πρώτη HTML φόρμα περνάει με τη μέθοδο POST τις χρονικές παραμέτρους που επέλεξε ο χρήστης στο δεύτερο php script. Σε αυτήν θα παρουσιάσουμε τους διαθέσιμους πόρους εκείνο το χρονικό διάστημα τους οποίους μπορεί να επιλέξει ο χρήστης για το slice του. Το δεύτερο php script κάνει έλεγχο αν ο χρήστης έχει δώσει τις κατάλληλες χρονικές παραμέτρους και αν η ημερομηνία δεν ανήκει στο παρελθόν. Αν κάτι από αυτά δεν ισχύει του εμφανίζει κατάλληλο μήνυμα και τον προτρέπει να ξαναεπιλέξει τις χρονικές παραμέτρους στο πρώτο php script. Σε αντίθετη περίπτωση του εμφανίζει τους διαθέσιμους πόρους για το χρονικό διάστημα που θέλει να χρησιμοποιήσει το καινούργιο του slice. Οι πόροι αυτοί περιέχονται στο RSpec v.2 advertisement που στέλνει η λειτουργία `getAvailableResources`, η οποία καλείται με php script στην συγκεκριμένη σελίδα. Χρησιμοποιώντας ένα XML parser παίρνουμε τις πληροφορίες και τις εμφανίζουμε στο χρήστη ώστε να επιλέξει τους διαθέσιμους πόρους που θέλει για το slice του. Παραθέτουμε παρακάτω μια εικόνα με την δεύτερη HTML φόρμα που ενημερώνει το χρήστη για τους διαθέσιμους πόρους:



Σχήμα 6.4: Επιλογή των ασύρματων πόρων που θα εκχωρηθούν στο slice

Κεφάλαιο 6: Περιγραφή Γραφικού Περιβάλλοντος Χρήστη (Graphical User Interface)

Από την παραπάνω εικόνα βλέπουμε ότι οι διαθέσιμοι κόμβοι για το καινούργιο slice είναι οι κόμβοι 4, 9 και 10. Επιπλέον σε HTML list παρέχονται και τα διαθέσιμα κανάλια μετάδοσης, ένα εκ των οποίων μπορεί ο χρήστης να ορίσει σε κάποιο interface του slice που δημιουργεί. Παραθέτουμε στην σελίδα αυτή και μια εικόνα της πειραματικής πλατφόρμας για να διευκολύνει το χρήστη να σχεδιάσει το slice του ανάλογα με την τοπολογία που επιθυμεί.

Η HTML φόρμα καλεί ως ACTION το τρίτο php script στο οποίο θα γίνει η τροποποίηση του slice και περνά τις παραμέτρους που επέλεξε ο χρήστης σε αυτό το php script με τη μέθοδο POST. Στο τρίτο php script αφού γίνει έλεγχος των παραμέτρων που έχει δώσει ο χρήστης για το slice που θέλει να δημιουργήσει, διαμορφώνεται ένα αντικείμενο τύπου RSpec, το οποίο περιέχει το όνομα του slice, το χρήστη του, την χρονική στιγμή έναρξης χρησιμοποίησης και τη χρονική διάρκεια χρησιμοποίησης. Επιπλέον το αντικείμενο RSpec έχει ως στοιχείο ένα RSpec v.2 Request, το οποίο διαμορφώνεται εντός του php script από τις παραμέτρους που δίνει ο χρήστης για τους πόρους και συμορφώνεται τόσο με το Core RSpec schema όσο και με το extension schema για τα χαρακτηριστικά των ασύρματων διεπαφών που θέλουμε. Αυτό το αντικείμενο RSpec στέλνεται ως παράμετρο στη λειτουργία mkslice της υπηρεσίας mksliceservice που αναλύσαμε στο Κεφάλαιο 5. Η υπηρεσία στέλνει ένα μήνυμα ότι πραγματοποιήθηκαν όλες οι διαδικασίες για την δημιουργία του συγκεκριμένου slice και ότι έχει κρατηθεί για το χρονικό διάστημα που επιλέξαμε στην πρώτη HTML φόρμα. Παρακάτω παραθέτουμε μια εικόνα για την HTML σελίδα, η οποία περιέχεται στο τρίτο php script στο οποίο ο χρήστης δημιουργεί το slice testslice9:

You made a slice

You made the slice testslice9

You added the node4 to the slice testslice9 with interfaces

- interface if0 with RadioChannel = 4 Rate = 6 OperatingMode = AP

You added the node10 to the slice testslice9 with interfaces

- interface if1 with RadioChannel = 8 Rate = 6 OperatingMode = AP

enripidis has reserved the slice testslice9 at 2011-07-03 05:00:00 for 2 hours.

[Return to the services](#)

Σχήμα 6.5: Επιβεβαίωση της δημιουργίας ενός καινούργιου slice με τους πόρους που επιλέχθηκαν

Παρατηρούμε από την παραπάνω εικόνα ότι ο χρήστης πρόσθεσε στο slice testslice9 το κόμβο 4 και το interface του if0 για το οποίο όρισε ρυθμό μετάδοσης 6 Mbps, κανάλι μετάδοσης το IEEE 802.11 b/g Channel 4 και τρόπο λειτουργίας ως Access Point, καθώς και το κόμβο 10 και το interface του if1 για το οποίο όρισε ρυθμό μετάδοσης 6 Mbps, κανάλι μετάδοσης το IEEE 802.11 b/g

Channel 8 και τρόπο λειτουργίας ως Access Point. Επιπλέον ο χρήστης κάνει reserve το slice για το συγκεκριμένο χρονικό διάστημα που επέλεξε.

Modify an existing slice-Τροποποίηση Υπάρχοντος slice του χρήστη

Η λειτουργία αυτή αποτελείται από τρία php scripts και χρησιμοποιείται για την τροποποίηση ενός slice του χρήστη. Στο πρώτο php script ενσωματώνεται μια HTML φόρμα και ο χρήστης καλείται να επιλέξει κάποιο από τα slices που έχει δημιουργήσει για να το τροποποιήσει προσθέτοντας ή αφαιρώντας κάποιους πόρους. Τα slices που μπορεί να επιλέξει ο χρήστης χωρίζονται σε δύο καταστάσεις στα No-activated και στα activated και τα No-activated με τη σειρά τους χωρίζονται σε άλλες δύο καταστάσεις. Αυτές είναι η No-activated and No-reserved slices, που περιλαμβάνει τα slices τα οποία ούτε έχουν δεσμευθεί για μελλοντική χρήση αλλά ούτε είναι ενεργοποιημένα αυτή τη χρονική στιγμή και η No-activated and reserved slices, που περιλαμβάνει τα slices τα οποία έχουν δεσμευθεί για μελλοντική χρήση αλλά δεν είναι ενεργοποιημένα αυτή τη χρονική στιγμή. Ακολουθούμε αυτήν την κατηγοριοποίηση ώστε ο χρήστης να ξέρει σε ποιά κατάσταση ανήκουν τα slices που έχει και να επιλέξει κατάλληλα αυτό που θέλει να τροποποιήσει. Για την αναπαράσταση σε λίστα των slices αυτών καλούνται διάφορες λειτουργίες της υπηρεσίας listslicesservice με παράμετρο το username του χρήστη εντός του php script. Ένα παράδειγμα της HTML φόρμας που έχει ενσωματωθεί στο πρώτο php script το οποίο καλεί τις διάφορες λειτουργίες για listing των slices και των στοιχείων που περιέχουν είναι:

← → ↻ 🏠 sorch.netmode.ntua.gr/ws/modifyslice.php

Αυτή η σελίδα είναι στα Αγγλικά. Θέλετε να τη μεταφράσετε; Μετάφραση Όχι Να μην γίνεται ποτέ μετάφραση από Αγγλικά

Modify a slice

No-activated slices

List of no-reserved Slices which are not activate now

- The testeur444 with owner evripidis contains the following nodes of testbed:
 1. The node8 with interfaces:
 - interface if0 with , Rate = 12 Mbps , RadioChannel = 1 , OperatingMode = AP
 2. The node10 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP
 - interface if1 with , Rate = 48 Mbps , RadioChannel = 11 , OperatingMode = STA

List of reserved Slices which are not activate now

- The test_slice23 with owner evripidis has been reserved from 2011-06-30 19:00:00 to 2011-06-30 21:00:00. The slice contains the following nodes of testbed:
 1. The node4 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP
 2. The node5 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP

Activated Slices

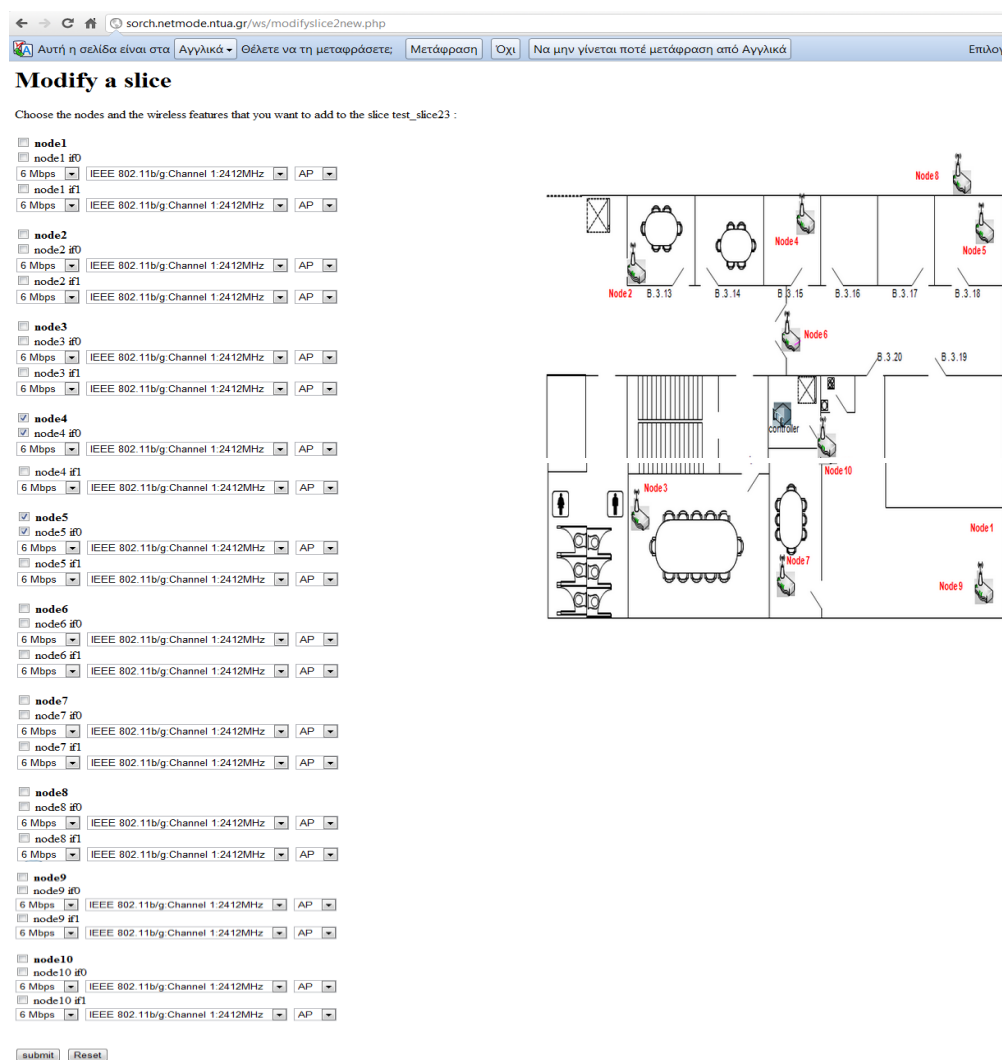
- The testslice9 with owner evripidis , which has been activated from 2011-06-30 17:00:00 and for 2 hours contains the following nodes of testbed:
 1. The node4 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 4 , OperatingMode = AP
 2. The node10 with interfaces:
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 8 , OperatingMode = AP

submit Reset

Σχήμα 6.6: Επιλογή του slice που θέλουμε να τροποποιήσουμε

Κεφάλαιο 6: Περιγραφή Γραφικού Περιβάλλοντος Χρήστη (Graphical User Interface)

Η HTML φόρμα αυτή καλεί ως ACTION το δεύτερο php script το οποίο ενσωματώνει μια καινούργια HTML φόρμα, στην οποία θα παρουσιάσουμε τους διαθέσιμους πόρους για το συγκεκριμένο slice. Η πρώτη HTML φόρμα περνάει με τη μέθοδο POST σαν παράμετρο στο δεύτερο php script το όνομα του συγκεκριμένου slice που επιλέχθηκε. Μέσα στο php script καλούνται οι λειτουργίες getListSliceResources και getListAvailableResources, οι οποίες επιστρέφουν η μεν πρώτη ένα RSpec v.2 Manifest σχετικά με τους πόρους που έχουν εκχωρηθεί στο slice και η μεν δεύτερη ένα RSpec v.2 Advertisement με τους διαθέσιμους πόρους για το συγκεκριμένο slice. Χρησιμοποιώντας κατάλληλο XML parser παίρνουμε τις πληροφορίες για το ποιούς πόρους έχει επιλέξει ο χρήστης και τι είναι διαθέσιμο να επιλέξει για να τροποποιήσει το συγκεκριμένο slice ώστε να το παρουσιάσουμε κατάλληλα στο user interface. Παραθέτουμε παρακάτω μια εικόνα με την δεύτερη HTML φόρμα που ενημερώνει το χρήστη για τους διαθέσιμους πόρους και για το ποιούς πόρους έχει ήδη επιλέξει στο συγκεκριμένο slice:



Σχήμα 6.7: Επιλογή των ασύρματων πόρων στο τροποποιημένο slice

Κεφάλαιο 6: Περιγραφή Γραφικού Περιβάλλοντος Χρήστη (Graphical User Interface)

Όπως βλέπουμε από την παραπάνω εικόνα το συγκεκριμένο slice περιέχει τον κόμβους 4 και 5, τα αντίστοιχα interfaces if0 του καθενός και τα αντίστοιχα χαρακτηριστικά των διεπαφών που έχουν επιλεγθεί από το χρήστη. Παράλληλα μας παρέχει και όλους τους υπόλοιπους διαθέσιμους κόμβους και τα διαθέσιμα κανάλια μετάδοσης που μπορούν να εκχωρηθούν στις ασύρματες διεπαφές. Παραθέτουμε στην σελίδα αυτή και μια εικόνα της πειραματικής πλατφόρμας για να διευκολύνει το χρήστη να τροποποιήσει το slice του ανάλογα με την τοπολογία που επιθυμεί.

Η δεύτερη HTML φόρμα καλεί ως ACTION το τρίτο php script στο οποίο θα γίνει η τροποποίηση του slice και περνάει της διάφορες παραμέτρους σε αυτό το script με τη μέθοδο POST. Στο τρίτο php script αφού γίνει έλεγχος των παραμέτρων που έχει δώσει ο χρήστης για το slice που θέλει να τροποποιήσει, διαμορφώνεται ένα αντικείμενο τύπου RSpec, το οποίο περιέχει το όνομα του slice καθώς και ένα RSpec v.2 Request, το οποίο διαμορφώνεται εντός του php script από τις παραμέτρους που δίνει ο χρήστης για τους πόρους και συμμορφώνεται τόσο με το Core RSpec schema όσο και με το extension schema για τα χαρακτηριστικά των ασύρματων διεπαφών που θέλουμε. Αυτό το αντικείμενο RSpec στέλνεται ως παράμετρος στη λειτουργία modifyslice της υπηρεσίας mksliceservice που αναλύσαμε στο Κεφάλαιο 5. Η υπηρεσία στέλνει ένα μήνυμα ότι πραγματοποιήθηκαν όλες οι διαδικασίες για την τροποποίηση του συγκεκριμένου slice εισάγοντας ή αφαιρώντας διαθέσιμους πόρους. Παρακάτω παραθέτουμε μια εικόνα για την HTML σελίδα που ενσωματώνεται στο php script στο οποίο ο χρήστης τροποποιεί το slice test_slice23:

You modified the slice test_slice23

You modified the slice test_slice23

The node4 to the slice test_slice23 with interfaces

- interface if0 with RadioChannel = 1 Rate = 6 OperatingMode = AP

The node5 to the slice test_slice23 with interfaces

- interface if0 with RadioChannel = 1 Rate = 6 OperatingMode = AP

The node7 to the slice test_slice23 with interfaces

- interface if0 with RadioChannel = 11 Rate = 12 OperatingMode = STA

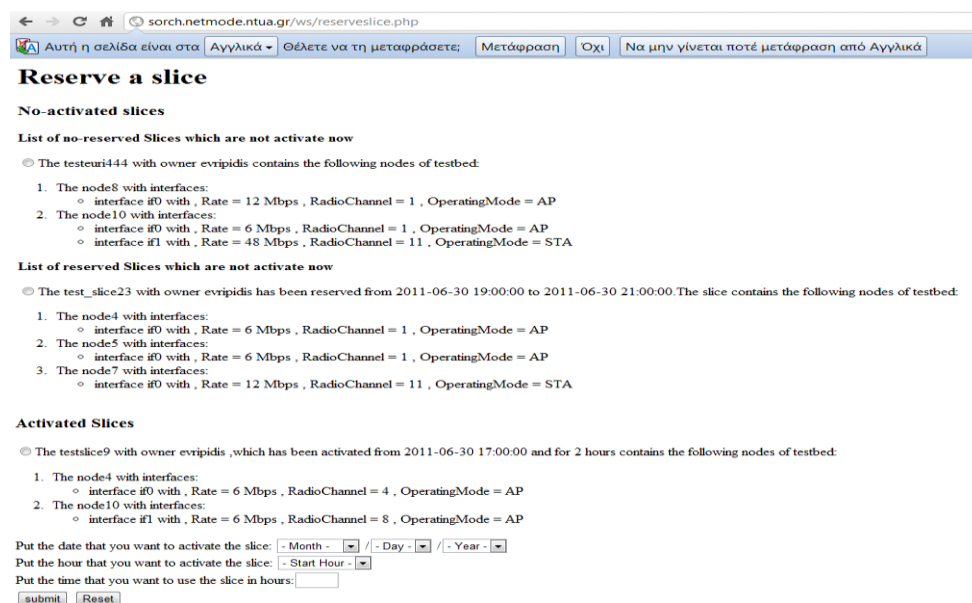
[Return to the services](#)

Σχήμα 6.8: Επιβεβαίωση της τροποποίησης ενός slice με τους πόρους που επιλέχθηκαν

Παρατηρούμε από την παραπάνω εικόνα ότι ο χρήστης διατήρησε στο slice του τους κόμβους 4 και 5 με τα αντίστοιχες διεπαφές και τα χαρακτηριστικά που είχε επιλέξει και πρόσθεσε επιπλέον τον κόμβο 7 με τη διεπαφή του if0 στην οποία όρισε κανάλι μετάδοσης το IEEE 802.11 b/g Channel 11 ,ρυθμό μετάδοσης 12 Mbps και τρόπο λειτουργίας ως Wireless Client.

Reserve a slice-Μελλοντική κράτηση ενός slice του χρήστη

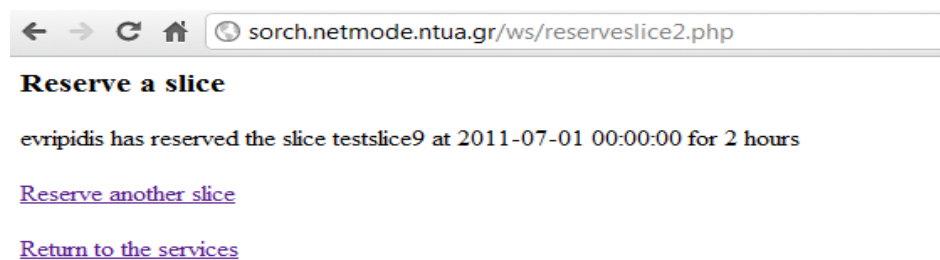
Η λειτουργία αυτή αποτελείται από δυο php scripts και χρησιμοποιείται για να κάνει κράτηση ο χρήστης εάν αυτό είναι δυνατό ενός συγκεκριμένου slice του για μελλοντική χρήση. Στο πρώτο php script ενσωματώνεται μια HTML φόρμα και ο χρήστης καλείται να επιλέξει το slice που θέλει να δεσμεύσει καθώς και την ημερομηνία, την ώρα έναρξης καθώς και τη διάρκεια χρησιμοποίησης του slice σε ώρες. Τα slices που μπορεί να επιλέξει ο χρήστης χωρίζονται σε δύο καταστάσεις στα No-activated και στα activated και τα No-activated με τη σειρά τους χωρίζονται σε άλλες δύο καταστάσεις. Αυτές είναι η No-activated and No-reserved slices, που περιλαμβάνει τα slices τα οποία ούτε έχουν δεσμευθεί για μελλοντική χρήση αλλά ούτε είναι ενεργοποιημένα αυτή τη χρονική στιγμή και η No-activated and reserved slices, που περιλαμβάνει τα slices τα οποία έχουν δεσμευθεί για μελλοντική χρήση αλλά δεν είναι ενεργοποιημένα αυτή τη χρονική στιγμή. Ακολουθούμε αυτήν την κατηγοριοποίηση ώστε ο χρήστης να ξέρει σε ποιά κατάσταση ανήκουν τα slices που έχει και να επιλέξει κατάλληλα αυτό που θέλει να δεσμεύσει. Για την αναπαράσταση σε λίστα των slices αυτών καλούνται διάφορες λειτουργίες της υπηρεσίας listslicesservice με παράμετρο το username του χρήστη εντός του php script. Ένα παράδειγμα της πρώτης HTML φόρμας που ενσωματώνεται στο php script το οποίο καλεί τις διάφορες λειτουργίες για listing των slices και των στοιχείων που περιέχουν είναι:



Σχήμα 6.9: Επιλογή του slice που θέλουμε να δεσμεύσουμε για μελλοντική χρήση και επιλογή και του χρονικού διαστήματος

Η HTML φόρμα καλεί ως ACTION το δεύτερο php script. Στο δεύτερο php script περνάνε ως μεταβλητές τόσο το όνομα του slice όσο και οι χρονικοί παράμετροι με τη μέθοδο POST μέσω της πρώτης HTML φόρμας. Γίνεται αρχικά

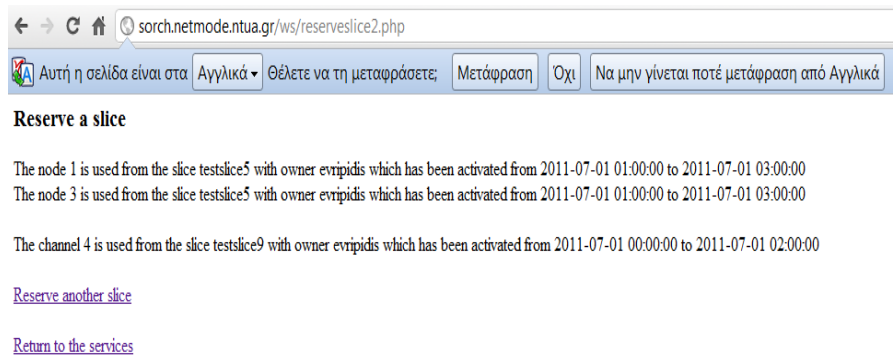
έλεγχος σχετικά με το αν ο χρήστης έδωσε τις απαραίτητες παραμέτρους και στη συνέχεια γίνεται και έλεγχος σχετικά με τις χρονικές παραμέτρους που εισήγαγε ο χρήστης. Αν η ημερομηνία και η ώρα που επέλεξε ανήκουν στο παρελθόν τότε του εμφανίζει μήνυμα που τον προτρέπει να προσπαθήσει να κάνει κράτηση κάποια άλλη ώρα. Στην περίπτωση που περάσει με επιτυχία τους δύο παραπάνω ελέγχους τότε το php script καλεί τη λειτουργία reserveslice της υπηρεσίας reservesliceservice με παραμέτρους το όνομα του slice και το χρονικό διάστημα που θέλουμε να το χρησιμοποιήσουμε προκειμένου να γίνει έλεγχος αν μπορεί να γίνει κράτηση του συγκεκριμένου slice για αυτό χρονικό διάστημα. Αν περάσει επιτυχώς τους ελέγχους της λειτουργίας αυτής για μη χρονική επικάλυψη χρήσης τόσο των κόμβων όσο και των καναλιών που ανήκουν στο συγκεκριμένο slice από κάποιο άλλο slice που έχει ενεργοποιηθεί σε εκείνη ή σε κάποια παρεμβάλλουσα χρονική περίοδο τότε θα επιστραφεί και θα τυπωθεί το μήνυμα ότι ο χρήστης έκανε επιτυχή κράτηση του συγκεκριμένου slice στην χρονική περίοδο που ήθελε. Ένα παράδειγμα επιτυχούς κράτησης κάποιου ήδη υπάρχοντος slice φαίνεται στην παρακάτω εικόνα όπου ο χρήστης ενριpidis έκανε κράτηση του slice του testslice9 για μια δεδομένη χρονική περίοδο:



Σχήμα 6.10: Επιβεβαίωση της κράτησης ενός ήδη υπάρχοντος slice για δεδομένο χρονικό διάστημα

Στην περίπτωση που δεν περάσει τους ελέγχους που γίνονται στο reservesliceservice τότε θα τυπωθεί ένα μήνυμα που θα ενημερώνει το χρήστη ποιοί πόροι (κόμβοι και κανάλια μετάδοσης) είναι δεσμευμένοι και από ποιο slice εκείνη τη χρονική περίοδο που έχει επιλέξει. Παρακάτω έχουμε το παράδειγμα ενός slice στο οποίο δεν μπορούμε να κάνουμε κράτηση μια δεδομένη χρονική περίοδο καθώς υπάρχει χρονική επικάλυψη χρήσης από άλλα slices δύο κόμβων και ενός καναλιού:

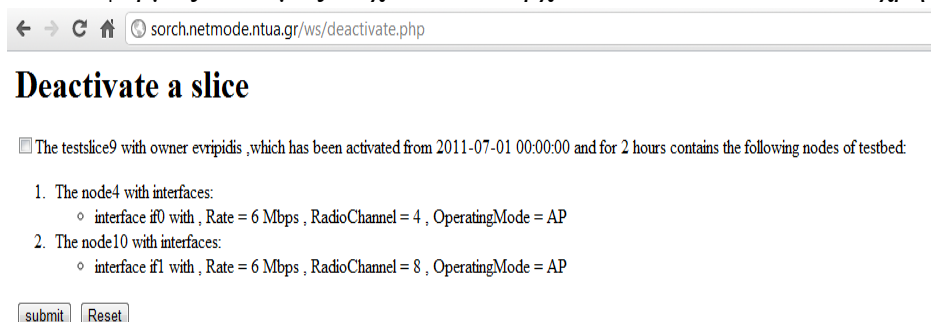
Κεφάλαιο 6: Περιγραφή Γραφικού Περιβάλλοντος Χρήστη (Graphical User Interface)



Σχήμα 6.11: Χρονική Επικάλυψη χρήσης πόρων (κόμβων ή καναλιών) από άλλα slice εκείνο το χρονικό διάστημα

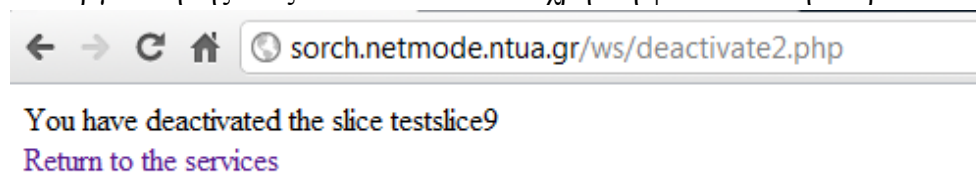
Deactivate slices-Απενεργοποίηση ενός ή περισσότερων activated slices του χρήστη

Η λειτουργία αυτή αποτελείται από δύο php scripts και χρησιμοποιείται για την απενεργοποίηση ενός ή περισσότερων activated slices του χρήστη. Στο πρώτο php script ενσωματώνεται μια HTML φόρμα και ο χρήστης καλείται να επιλέξει ένα ή περισσότερα activated slices του ώστε να τα απενεργοποιήσει. Η σελίδα αυτή περιέχει μια λίστα των activated slices την οποία παίρνουμε καλώντας στο php script τη λειτουργία getMyActivatedSlices της υπηρεσίας listslicesservice. Αυτή η λειτουργία μας επιστρέφει ένα αντικείμενο testbedslices που περιέχει ένα πίνακα από αντικείμενα τύπου slice τα οποία περιέχουν για κάθε activated slice του χρήστη το όνομα του slice, το όνομα του χρήστη, τη χρονική στιγμή έναρξης ενεργοποίησης, τη χρονική διάρκεια ενεργοποίησης και ένα RSpec v.2 Manifest που περιλαμβάνει τους πόρους που έχουν εκχωρηθεί στο slice με τα αντίστοιχα χαρακτηριστικά τους. Χρησιμοποιώντας XML parser μπορούμε να πάρουμε τα διάφορα στοιχεία του κάθε activated slice και να τα παρουσιάσουμε στο χρήστη. Παραθέτουμε μια εικόνα της HTML φόρμας όπου μας δείχνει ότι υπάρχει ένα activated slice του χρήστη:



Σχήμα 6.12: Επιλογή ενός ή περισσότερων ενεργοποιημένων (activated) slices του χρήστη για απενεργοποίηση

Η HTML φόρμα καλεί ως ACTION το δεύτερο php script. Στο δεύτερο php script περνάει ένα ή περισσότερα ονόματα activated slices του χρήστη που ο χρήστης θέλει να απενεργοποιήσει με τη μέθοδο POST μέσω της πρώτης HTML φόρμας. Το δεύτερο php script καλεί τη λειτουργία deactivateslice της υπηρεσίας deactivateservice με παράμετρο κάθε ένα από τα ονόματα των activated slices που θέλει να απενεργοποιήσει. Αν πραγματοποιηθούν επιτυχώς οι διαδικασίες της λειτουργίας της υπηρεσίας τότε θα επιστραφεί και θα τυπωθεί το μήνυμα ότι ο χρήστης έκανε επιτυχή απενεργοποίηση του συγκεκριμένου slice. Αυτό συμβαίνει για κάθε activated slice που έχει επιλέξει να απενεργοποιήσει. Ένα παράδειγμα επιτυχούς απενεργοποίησης ενός activated slice του χρήστη φαίνεται στην παρακάτω εικόνα:



Σχήμα 6.13: Επιβεβαίωση επιτυχούς απενεργοποίησης ενός slice

Cancel Reservations-Ακύρωση ενός ή περισσότερων κρατήσεων (reservations) των slices του χρήστη

Η λειτουργία αυτή αποτελείται από δύο php scripts και χρησιμοποιείται για την ακύρωση ενός ή περισσότερων κρατήσεων των slices του χρήστη. Στο πρώτο php script ενσωματώνεται μια HTML φόρμα και ο χρήστης καλείται να επιλέξει ένα ή περισσότερα reservations των reserved slices ώστε να τα ακυρώσει. Η σελίδα αυτή περιέχει μια λίστα των reserved slices του χρήστη την οποία παίρνουμε καλώντας στο php script τη λειτουργία getMyReservedSlices της υπηρεσίας listslicesservice, καθώς και μια λίστα με τα διάφορα reservations κάτω από κάθε reserved slice. Αυτό συμβαίνει γιατί κάθε reserved slice του χρήστη μπορεί να έχει κρατηθεί για ένα ή περισσότερα χρονικά διαστήματα στο μέλλον. Η λειτουργία getMyReservedSlices δέχεται ως παράμετρο το username του χρήστη και μας επιστρέφει ένα αντικείμενο testbedslices που περιέχει ένα πίνακα από αντικείμενα τύπου slice τα οποία περιέχουν για κάθε activated slice του χρήστη το όνομα του slice στο στοιχείο \$name, το όνομα του χρήστη στο στοιχείο \$owner, τις χρονικές στιγμές έναρξης της ενεργοποίησης του στο στοιχείο \$time_start, τις χρονικές στιγμές λήξης της ενεργοποίησης του στο στοιχείο \$time_end και ένα RSpec v.2 Manifest που περιλαμβάνει τους πόρους που έχουν εκχωρηθεί στο slice με τα αντίστοιχα χαρακτηριστικά τους στο στοιχείο \$rspec. Χρησιμοποιώντας XML parser μπορούμε να πάρουμε τα διάφορα στοιχεία του κάθε reserved slice και να τα παρουσιάσουμε στο χρήστη. Επιπλέον, από τα στοιχεία \$time_start και \$time_end των αντικειμένων slice μπορούμε να πάρουμε τις χρονικές στιγμές έναρξης και λήξης της ενεργοποίησης του κάθε reserved slice ώστε να δημιουργήσουμε τη λίστα με τις κρατήσεις κάτω από κάθε reserved slice. Παραθέτουμε μια εικόνα της HTML φόρμας όπου μας δείχνει ότι υπάρχει ένα reserved slice του χρήστη, στο οποίο έχει γίνει κράτηση για δύο χρονικά διαστήματα:

Κεφάλαιο 6:Περιγραφή Γραφικού Περιβάλλοντος Χρήστη (Graphical User Interface)

← → ↻ 🏠 🌐 sorch.netmode.ntua.gr/ws/reservationcancel.php

List of Reservations

- The testslice5 with owner evripdis has been reserved from 2011-07-01 10:00:00 to 2011-07-01 12:00:00 and from 2011-07-01 13:00:00 to 2011-07-01 15:00:00.The slice contains the following of testbed:
 1. The node1 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 2. The node3 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP

Reservations

from 2011-07-01 10:00:00 to 2011-07-01 12:00:00

from 2011-07-01 13:00:00 to 2011-07-01 15:00:00

Σχήμα 6.14: Επιλογή ενός ή περισσότερων κρατήσεων του χρήστη για ακύρωση

Η HTML φόρμα καλεί ως ACTION το δεύτερο php script. Στο δεύτερο php script περνάει ένα ή περισσότερα ονόματα reserved slices του χρήστη και μαζί με κάθε ένα από αυτά την χρονική στιγμή έναρξης της κράτησης που ο χρήστης θέλει να ακυρώσει, με τη μέθοδο POST μέσω της πρώτης HTML φόρμας. Το δεύτερο php script καλεί τη λειτουργία cancelreservation της υπηρεσίας cancelreservationservice με παράμετρο κάθε έναν από αυτούς του συνδυασμούς ονόματος reserved slice και χρονικής στιγμής έναρξης της κράτησης του ανάλογα με το ποιές κρατήσεις θέλει να απενεργοποιήσει. Αν πραγματοποιηθούν επιτυχώς οι διαδικασίες της λειτουργίας της υπηρεσίας τότε θα επιστραφεί και θα τυπωθεί το μήνυμα ότι ο χρήστης έκανε επιτυχή ακύρωση της συγκεκριμένης κράτησης. Αυτό συμβαίνει για κάθε κράτηση που έχει επιλέξει να ακυρώσει. Ένα παράδειγμα επιτυχούς ακύρωσης μιας κράτησης ενός reserved slice του χρήστη φαίνεται στην παρακάτω εικόνα:

← → ↻ 🏠 🌐 sorch.netmode.ntua.gr/ws/reservationcancel2.php

You have cancelled the reservation of the slice testslice5 2011-07-01 13:00:00

[Return to the services](#)

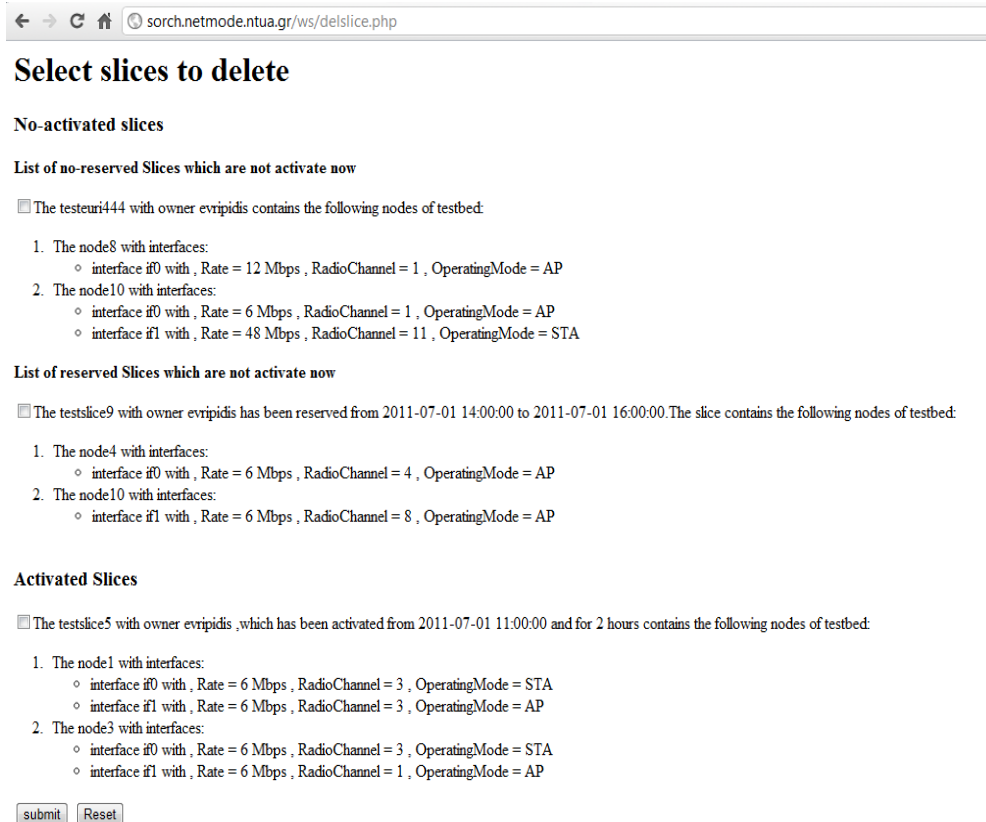
Σχήμα 6.15: Επιβεβαίωση ακύρωσης μιας κράτησης για κάποιο slice του χρήστη

Delete slices-Διαγραφή ενός ή περισσότερων slices του χρήστη

Η λειτουργία αυτή αποτελείται από δύο php scripts και χρησιμοποιείται για την διαγραφή ενός ή περισσότερων slices του χρήστη. Στο πρώτο php script ενσωματώνεται μια HTML φόρμα και ο χρήστης καλείται να επιλέξει ένα ή περισσότερα slices που έχει δημιουργήσει προκειμένου να τα διαγράψει. Τα slices που μπορεί να επιλέξει ο χρήστης χωρίζονται σε δύο καταστάσεις στα No-activated και στα activated και τα No-activated με τη σειρά τους χωρίζονται σε άλλες δύο καταστάσεις. Αυτές είναι η No-activated and No-reserved slices, που περιλαμβάνει τα slices τα οποία ούτε έχουν δεσμευθεί για μελλοντική χρήση αλλά ούτε είναι

Κεφάλαιο 6: Περιγραφή Γραφικού Περιβάλλοντος Χρήστη (Graphical User Interface)

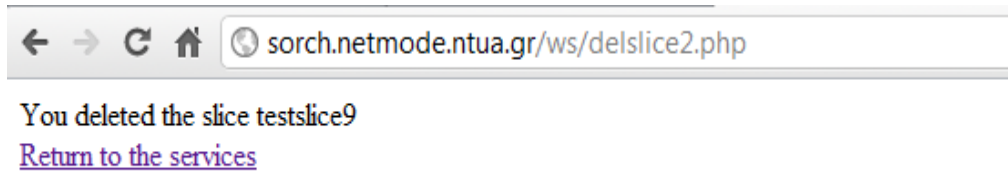
ενεργοποιημένα αυτή τη χρονική στιγμή και η No-activated and reserved slices, που περιλαμβάνει τα slices τα οποία έχουν δεσμευθεί για μελλοντική χρήση αλλά δεν είναι ενεργοποιημένα αυτή τη χρονική στιγμή. Ακολουθούμε αυτήν την κατηγοριοποίηση ώστε ο χρήστης να ξέρει σε ποιά κατάσταση ανήκουν τα slices που έχει και να επιλέξει κατάλληλα αυτά που θέλει να διαγράψει. Για την αναπαράσταση σε λίστα των slices αυτών καλούνται διάφορες λειτουργίες της υπηρεσίας listslicesservice με παράμετρο το username του χρήστη εντός του php script. Ένα παράδειγμα της HTML φόρμας που έχει ενσωματωθεί στο πρώτο php script το οποίο καλεί τις διάφορες λειτουργίες για listing των slices και των στοιχείων που περιέχουν είναι:



The screenshot shows a web browser window with the address bar containing 'sorch.netmode.ntua.gr/ws/delslice.php'. The page title is 'Select slices to delete'. Under the heading 'No-activated slices', there is a sub-heading 'List of no-reserved Slices which are not activate now'. A checkbox is followed by the text 'The testeur444 with owner evripidis contains the following nodes of testbed:'. Below this, there are two numbered lists of nodes with their interfaces and configurations. The first list includes 'The node8 with interfaces' and 'The node10 with interfaces'. The second list includes 'The node4 with interfaces' and 'The node10 with interfaces'. Under the heading 'Activated Slices', there is a sub-heading 'List of reserved Slices which are not activate now'. A checkbox is followed by the text 'The testslice9 with owner evripidis has been reserved from 2011-07-01 14:00:00 to 2011-07-01 16:00:00. The slice contains the following nodes of testbed:'. Below this, there are two numbered lists of nodes with their interfaces and configurations. The first list includes 'The node1 with interfaces' and 'The node3 with interfaces'. At the bottom of the page, there are two buttons: 'submit' and 'Reset'.

Σχήμα 6.16: Επιλογή ενός ή περισσότερων slices του χρήστη για διαγραφή

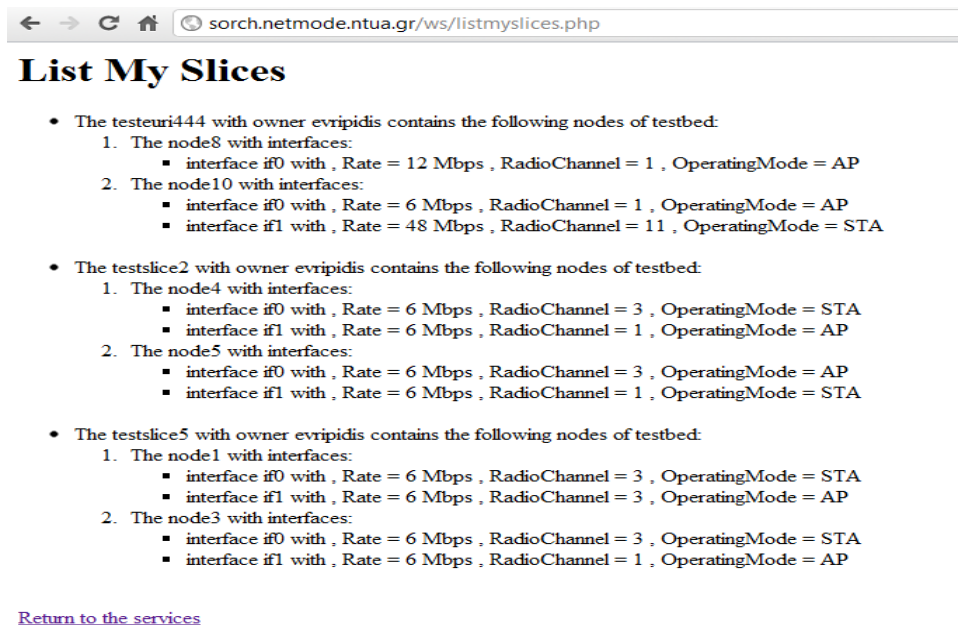
Η HTML φόρμα καλεί ως ACTION το δεύτερο php script. Στο δεύτερο php script περνάει ένα ή περισσότερα ονόματα slices του χρήστη που ο χρήστης θέλει να διαγράψει με τη μέθοδο POST μέσω της πρώτης HTML φόρμας. Το δεύτερο php script καλεί τη λειτουργία delslice της υπηρεσίας delslliceservice με παράμετρο κάθε ένα από τα ονόματα των slices που θέλει να διαγράψει. Αν πραγματοποιηθούν επιτυχώς οι διαδικασίες της λειτουργίας της υπηρεσίας τότε θα επιστραφεί και θα τυπωθεί το μήνυμα ότι ο χρήστης έκανε επιτυχή διαγραφή του συγκεκριμένου slice. Αυτό συμβαίνει για κάθε slice που έχει επιλέξει να διαγράψει. Ένα παράδειγμα επιτυχούς διαγραφής ενός slice του χρήστη φαίνεται στην παρακάτω εικόνα:



Σχήμα 6.17: Επιβεβαίωση διαγραφής ενός slice

List my slices-Λίστα των slices του συγκεκριμένου χρήστη

Η λειτουργία αποτελείται από ένα php script στο οποίο ενσωματώνουμε μια HTML σελίδα και χρησιμοποιείται για να παρουσιάσει στο χρήστη σε μορφή λίστας το σύνολο των slices που έχει δημιουργήσει. Η σελίδα αυτή περιέχει μια λίστα των slices που ανήκουν στο χρήστη την οποία παίρνουμε καλώντας στο php script τη λειτουργία `getMySlices` της υπηρεσίας `listslicesservice` με παράμετρο το `username` του χρήστη. Αυτή η λειτουργία μας επιστρέφει ένα αντικείμενο `testbedslices` που περιέχει ένα πίνακα από αντικείμενα τύπου `slice` τα οποία περιέχουν για κάθε slice του χρήστη το όνομα του, το όνομα του χρήστη και ένα `RSpec v.2 Manifest` που περιλαμβάνει τους πόρους που έχουν εκχωρηθεί στο slice με τα αντίστοιχα χαρακτηριστικά τους. Χρησιμοποιώντας XML parser μπορούμε να πάρουμε τα διάφορα στοιχεία του κάθε slice και να τα παρουσιάσουμε στο χρήστη. Παραθέτουμε μια εικόνα της HTML σελίδας όπου μας δείχνει ότι ο χρήστης `evripidis` έχει δημιουργήσει τρία slices:



Σχήμα 6.18: Λίστα των slices του συγκεκριμένου χρήστη

List my no-reserved slices-Λίστα των no-reserved slices του συγκεκριμένου χρήστη

Η λειτουργία αποτελείται από ένα php script στο οποίο ενσωματώνουμε μια HTML σελίδα και χρησιμοποιείται για να παρουσιάσει στο χρήστη σε μορφή λίστας το σύνολο των no-reserved slices που του ανήκουν. Η σελίδα αυτή περιέχει μια λίστα των no-reserved slices που ανήκουν στο χρήστη την οποία παίρνουμε καλώντας στο php script τη λειτουργία getMyNoReservedSlices της υπηρεσίας listslicesservice με παράμετρο το username του χρήστη. Αυτή η λειτουργία μας επιστρέφει ένα αντικείμενο testbedslices που περιέχει ένα πίνακα από αντικείμενα τύπου slice τα οποία περιέχουν για κάθε slice του χρήστη το όνομα του, το όνομα του χρήστη και ένα RSpec v.2 Manifest που περιλαμβάνει τους πόρους που έχουν εκχωρηθεί στο no-reserved slice με τα αντίστοιχα χαρακτηριστικά τους. Χρησιμοποιώντας XML parser μπορούμε να πάρουμε τα διάφορα στοιχεία του κάθε no-reserved slice και να τα παρουσιάσουμε στο χρήστη. Παραθέτουμε μια εικόνα της HTML σελίδας όπου μας δείχνει ότι ο χρήστης elena έχει δύο no-reserved slices:

List of no-reserved Slices

- The testslice8000 with owner elena contains the following nodes of testbed:
 1. The node6 with interfaces:
 - interface if1 with , Rate = 11 Mbps , RadioChannel = 1 , OperatingMode = AP
 2. The node10 with interfaces:
 - interface if1 with , Rate = 11 Mbps , RadioChannel = 1 , OperatingMode = STA
- The testelena6 with owner elena contains the following nodes of testbed:
 1. The node4 with interfaces:
 - interface if0 with , Rate = 2 Mbps , RadioChannel = 3 , OperatingMode = AP
 - interface if1 with , Rate = 1 Mbps , RadioChannel = 3 , OperatingMode = AP
 2. The node5 with interfaces:
 - interface if0 with , Rate = 1 Mbps , RadioChannel = 3 , OperatingMode = AP
 - interface if1 with , Rate = 1 Mbps , RadioChannel = 3 , OperatingMode = AP

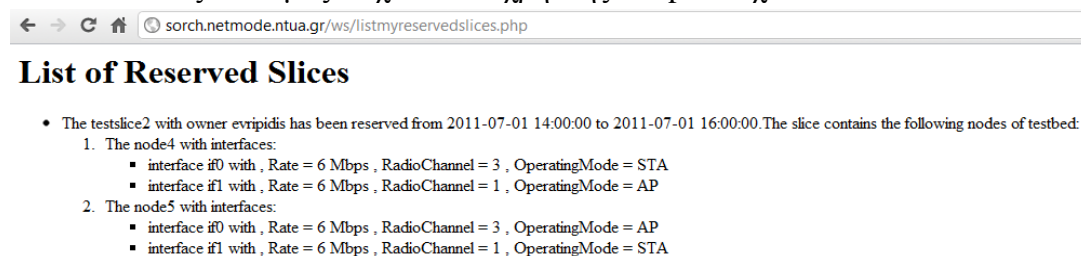
[Return to the services](#)

Σχήμα 6.19: Λίστα των no-reserved slices του συγκεκριμένου χρήστη

List my reserved slices-Λίστα των reserved slices του συγκεκριμένου χρήστη

Η λειτουργία αποτελείται από ένα php script στο οποίο ενσωματώνουμε μια HTML σελίδα και χρησιμοποιείται για να παρουσιάσει στο χρήστη σε μορφή λίστας το σύνολο των reserved slices που του ανήκουν. Η σελίδα αυτή περιέχει μια λίστα των reserved slices που ανήκουν στο χρήστη την οποία παίρνουμε καλώντας στο php script τη λειτουργία getMyReservedSlices της υπηρεσίας listslicesservice με παράμετρο το username του χρήστη. Αυτή η λειτουργία μας επιστρέφει ένα αντικείμενο testbedslices που περιέχει ένα πίνακα από αντικείμενα τύπου slice τα οποία περιέχουν για κάθε slice του χρήστη το όνομα του, το όνομα του χρήστη, την χρονική στιγμή έναρξης της ενεργοποίησης του slice, τη χρονική στιγμή λήξης της ενεργοποίησης του slice και ένα RSpec v.2 Manifest που περιλαμβάνει τους πόρους που έχουν εκχωρηθεί στο reserved slice με τα αντίστοιχα χαρακτηριστικά τους.

Χρησιμοποιώντας XML parser μπορούμε να πάρουμε τα διάφορα στοιχεία του κάθε reserved slice και να τα παρουσιάσουμε στο χρήστη. Παραθέτουμε μια εικόνα της HTML σελίδας όπου μας δείχνει ότι ο χρήστης enripidis έχει ένα reserved slice:



← → ↻ 🏠 sorch.netmode.ntua.gr/ws/listmyreservedslices.php

List of Reserved Slices

- The testslice2 with owner enripidis has been reserved from 2011-07-01 14:00:00 to 2011-07-01 16:00:00. The slice contains the following nodes of testbed:
 1. The node4 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP
 2. The node5 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = STA

[Return to the services](#)

Σχήμα 6.20: Λίστα των reserved slices του συγκεκριμένου χρήστη

List my no-activated slices-Λίστα των no-activated slices του συγκεκριμένου χρήστη

Η λειτουργία αποτελείται από ένα php script στο οποίο ενσωματώνουμε μια HTML σελίδα και χρησιμοποιείται για να παρουσιάσει στο χρήστη σε μορφή λίστας το σύνολο των no-activated slices που του ανήκουν. Τα no-activated slices τα χωρίζουμε σε δύο καταστάσεις στις No-activated and No-reserved slices και No-activated and reserved slices, όπως αναφέραμε στο υποκεφάλαιο 5.1. Στην κατάσταση No-activated and No-reserved slices ανήκουν τα slices του χρήστη που ούτε είναι ενεργοποιημένα αυτή τη χρονική στιγμή αλλά ούτε έχουν δεσμευθεί για μελλοντική χρήση. Στην κατάσταση No-activated and reserved slices ανήκουν τα slices τα οποία δεν είναι ενεργοποιημένα αυτή τη χρονική στιγμή αλλά έχουν δεσμευθεί για μια ή περισσότερες χρονικές περιόδους στο μέλλον. Ακολουθούμε αυτήν την κατηγοριοποίηση ώστε ο χρήστης να ξέρει σε ποιά επιμέρους κατάσταση ανήκουν τα no-activated slices. Για την αναπαράσταση σε λίστα των slices αυτών καλούνται διάφορες λειτουργίες της υπηρεσίας listslicesservice με παράμετρο το username του χρήστη εντός του php script. Παραθέτουμε μια εικόνα της HTML σελίδας όπου μας δείχνει ότι ο χρήστης enripidis έχει τρία no-activated slices, εκ των οποίων τα δύο ανήκουν στην κατάσταση No-activated and No-reserved που περιγράψαμε και το ένα στην κατάσταση No-activated and reserved slices:

List of no-activated Slices

List of no-reserved Slices which are not activate now

- The testeur444 with owner evripidis contains the following nodes of testbed:
 1. The node8 with interfaces:
 - interface #0 with , Rate = 12 Mbps , RadioChannel = 1 , OperatingMode = AP
 2. The node10 with interfaces:
 - interface #0 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP
 - interface #1 with , Rate = 48 Mbps , RadioChannel = 11 , OperatingMode = STA
- The testslice5 with owner evripidis contains the following nodes of testbed:
 1. The node1 with interfaces:
 - interface #0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface #1 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 2. The node3 with interfaces:
 - interface #0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface #1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP

List of reserved Slices which are not activate now

- The testslice2 with owner evripidis has been reserved from 2011-07-01 14:00:00 to 2011-07-01 16:00:00. The slice contains the following nodes of testbed:
 1. The node4 with interfaces:
 - interface #0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface #1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP
 2. The node5 with interfaces:
 - interface #0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 - interface #1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = STA

[Return to the services](#)

Σχήμα 6.21: Λίστα των no-activated slices του συγκεκριμένου χρήστη

List my activated slices-Λίστα των activated slices του συγκεκριμένου χρήστη

Η λειτουργία αποτελείται απ;ο ένα php script στο οποίο ενσωματώνουμε μια HTML σελίδα και χρησιμοποιείται για να παρουσιάσει στο χρήστη σε μορφή λίστας το σύνολο των activated slices που του ανήκουν. Η σελίδα αυτή περιέχει μια λίστα των activated slices που ανήκουν στο χρήστη την οποία παίρνουμε καλώντας στο php script τη λειτουργία getMyActivatedSlices της υπηρεσίας listslicesservice με παράμετρο το username του χρήστη. Αυτή η λειτουργία μας επιστρέφει ένα αντικείμενο testbedslices που περιέχει ένα πίνακα από αντικείμενα τύπου slice τα οποία περιέχουν για κάθε slice του χρήστη το όνομα του, το όνομα του χρήστη, την χρονική στιγμή έναρξης της ενεργοποίησης του slice, τη χρονική διάρκεια ενεργοποίησης του slice και ένα RSpec v.2 Manifest που περιλαμβάνει τους πόρους που έχουν εκχωρηθεί στο activated slice με τα αντίστοιχα χαρακτηριστικά τους. Χρησιμοποιώντας XML parser μπορούμε να πάρουμε τα διάφορα στοιχεία του κάθε activated slice και να τα παρουσιάσουμε στο χρήστη. Παραθέτουμε μια εικόνα της HTML σελίδας όπου μας δείχνει ότι ο χρήστης evripidis έχει ένα activated slice:

← → ↻ 🏠 sorch.netmode.ntua.gr/ws/listmyactivatedslices.php

List of Activated Slices

- The testslice5 with owner evripidis ,which has been activated from 2011-07-01 11:00:00 and for 2 hours contains the following nodes of testbed:
 1. The node1 with interfaces:
 - interface #0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface #1 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 2. The node3 with interfaces:
 - interface #0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface #1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP

[Return to the services](#)

Σχήμα 6.22: Λίστα των activated slices του συγκεκριμένου χρήστη

List slices-Λίστα όλων των slices

Η λειτουργία αποτελείται από ένα php script στο οποίο ενσωματώνουμε μια HTML σελίδα και χρησιμοποιείται για να παρουσιάσει στο χρήστη σε μορφή λίστας το σύνολο των slices που έχουν δημιουργηθεί ανεξαρτήτως χρήστη. Το php script περιλαμβάνει διαδικασίες ίδιες με του php script στη λειτουργία List my slices με τη διαφορά ότι θέλουμε λίστα όλων των slices. Για αυτό το λόγο καλούμε τη λειτουργία listslices της υπηρεσίας listslicesservice χωρίς παράμετρο και στη συνέχεια ακολουθούμε την ίδια διαδικασία με το script στο List my slices. Παραθέτουμε μια εικόνα της HTML σελίδας όπου μας δείχνει ότι συνολικά υπάρχουν έξι slices που ανήκουν σε διαφορετικούς χρήστες:

List of Slices

- The elena1 with owner elena contains the following nodes of testbed:
 1. The node3 with interfaces:
 - interface if0 with , Rate = 18 Mbps , RadioChannel = 3 , OperatingMode = STA
 2. The node4 with interfaces:
 - interface if1 with , Rate = 18 Mbps , RadioChannel = 1 , OperatingMode = AP
- The testeur1444 with owner evripidis contains the following nodes of testbed:
 1. The node8 with interfaces:
 - interface if0 with , Rate = 12 Mbps , RadioChannel = 1 , OperatingMode = AP
 2. The node10 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP
 - interface if1 with , Rate = 48 Mbps , RadioChannel = 11 , OperatingMode = STA
- The testelena6565 with owner elena contains the following nodes of testbed:
 1. The node3 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 2. The node5 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 3. The node7 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
- The testslice2 with owner evripidis contains the following nodes of testbed:
 1. The node4 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP
 2. The node5 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = STA
- The testslice5 with owner evripidis contains the following nodes of testbed:
 1. The node1 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 2. The node3 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP
- The testeur17 with owner euri contains the following nodes of testbed:
 1. The node1 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 4 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 2 , OperatingMode = AP
 2. The node3 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 2 , OperatingMode = STA
 - interface if1 with , Rate = 12 Mbps , RadioChannel = 6 , OperatingMode = STA

[Return to the services](#)

Σχήμα 6.23: Λίστα όλων των slices που έχουν δημιουργήσει οι χρήστες

List no-reserved slices-Λίστα όλων των no-reserved slices

Η λειτουργία αποτελείται από ένα php script στο οποίο ενσωματώνουμε μια HTML σελίδα και χρησιμοποιείται για να παρουσιάσει στο χρήστη σε μορφή λίστας το σύνολο των no-reserved slices που υπάρχουν στην πειραματική πλατφόρμα. Η σελίδα αυτή περιέχει μια λίστα των no-reserved slices την οποία παίρνουμε

καλώντας στο php script τη λειτουργία `getNoReservedSlices` της υπηρεσίας `listslicesservice` με παράμετρο το `username` του χρήστη. Αυτή η λειτουργία μας επιστρέφει ένα αντικείμενο `testbedslices` που περιέχει ένα πίνακα από αντικείμενα τύπου `slice` τα οποία περιέχουν για κάθε `slice` του χρήστη το όνομα του, το όνομα του χρήστη και ένα `RSpec v.2 Manifest` που περιλαμβάνει τους πόρους που έχουν εκχωρηθεί στο `no-reserved slice` με τα αντίστοιχα χαρακτηριστικά τους. Χρησιμοποιώντας `XML parser` μπορούμε να πάρουμε τα διάφορα στοιχεία του κάθε `no-reserved slice` και να τα παρουσιάσουμε στο χρήστη. Παραθέτουμε μια εικόνα της `HTML` σελίδας όπου μας δείχνει ότι υπάρχουν τέσσερα `no-reserved slices` στην πειραματική πλατφόρμα:

List of no-reserved Slices

- The `gfdsgs` with owner `evripidis` contains the following nodes of testbed:
 1. The `node6` with interfaces:
 - `interface if1` with , `Rate = 11 Mbps` , `RadioChannel = 1` , `OperatingMode = AP`
 2. The `node10` with interfaces:
 - `interface if1` with , `Rate = 11 Mbps` , `RadioChannel = 1` , `OperatingMode = STA`
- The `testelena6` with owner `elena` contains the following nodes of testbed:
 1. The `node4` with interfaces:
 - `interface if0` with , `Rate = 2 Mbps` , `RadioChannel = 3` , `OperatingMode = AP`
 - `interface if1` with , `Rate = 1 Mbps` , `RadioChannel = 3` , `OperatingMode = AP`
 2. The `node5` with interfaces:
 - `interface if0` with , `Rate = 1 Mbps` , `RadioChannel = 3` , `OperatingMode = AP`
 - `interface if1` with , `Rate = 1 Mbps` , `RadioChannel = 3` , `OperatingMode = AP`
- The `testslice` with owner `evripidis` contains the following nodes of testbed:
 1. The `node2` with interfaces:
 - `interface if0` with , `Rate = 6 Mbps` , `RadioChannel = 4` , `OperatingMode = AP`
 2. The `node4` with interfaces:
 - `interface if0` with , `Rate = 6 Mbps` , `RadioChannel = 4` , `OperatingMode = AP`
- The `testeuri7` with owner `huri` contains the following nodes of testbed:
 1. The `node1` with interfaces:
 - `interface if0` with , `Rate = 5.5 Mbps` , `RadioChannel = 4` , `OperatingMode = STA`
 - `interface if1` with , `Rate = 6 Mbps` , `RadioChannel = 2` , `OperatingMode = AP`
 2. The `node3` with interfaces:
 - `interface if0` with , `Rate = 6 Mbps` , `RadioChannel = 2` , `OperatingMode = STA`
 - `interface if1` with , `Rate = 12 Mbps` , `RadioChannel = 6` , `OperatingMode = STA`

[Return to the services](#)

Σχήμα 6.24: Λίστα όλων των `no-reserved slices`

List reserved slices-Λίστα όλων των reserved slices

Η λειτουργία αποτελείται από ένα php script στο οποίο ενσωματώνουμε μια `HTML` σελίδα και χρησιμοποιείται για να παρουσιάσει στο χρήστη σε μορφή λίστας το σύνολο των `reserved slices` ανεξαρτήτως χρήστη. Το php script περιλαμβάνει διαδικασίες ίδιες με του php script στη λειτουργία `List my reserved slices` με τη διαφορά ότι θέλουμε λίστα όλων των `reserved slices`. Για αυτό το λόγο καλούμε τη λειτουργία `listreservedslices` της υπηρεσίας `listslicesservice` χωρίς

Κεφάλαιο 6: Περιγραφή Γραφικού Περιβάλλοντος Χρήστη (Graphical User Interface)

παράμετρο και στη συνέχεια ακολουθούμε την ίδια διαδικασία με το script στο List my reserved slices. Παραθέτουμε μια εικόνα της HTML σελίδας όπου μας δείχνει ότι συνολικά υπάρχουν τρία reserved slices για μελλοντική χρήση που ανήκουν σε διαφορετικούς χρήστες:

List of Reserved Slices

- The testeur1444 with owner evripidis has been reserved from 2011-07-01 16:00:00 to 2011-07-01 18:00:00. The slice contains the following nodes of testbed:
 1. The node8 with interfaces:
 - interface if0 with , Rate = 12 Mbps , RadioChannel = 1 , OperatingMode = AP
 2. The node10 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP
 - interface if1 with , Rate = 48 Mbps , RadioChannel = 11 , OperatingMode = STA
- The testeur17 with owner euri has been reserved from 2011-07-01 21:00:00 to 2011-07-01 23:00:00. The slice contains the following nodes of testbed:
 1. The node1 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 4 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 2 , OperatingMode = AP
 2. The node3 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 2 , OperatingMode = STA
 - interface if1 with , Rate = 12 Mbps , RadioChannel = 6 , OperatingMode = STA
- The testslice5 with owner evripidis has been reserved from 2011-07-01 18:00:00 to 2011-07-01 20:00:00. The slice contains the following nodes of testbed:
 1. The node1 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 2. The node3 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP

[Return to the services](#)

Σχήμα 6.25: Λίστα όλων των reserved slices

List no-activated slices-Λίστα όλων των no-activated slices

Η λειτουργία αποτελείται από ένα php script στο οποίο ενσωματώνουμε μια HTML σελίδα και χρησιμοποιείται για να παρουσιάσει στο χρήστη σε μορφή λίστας το σύνολο των no-activated slices ανεξαρτήτως χρήστη. Το php script περιλαμβάνει διαδικασίες ίδιες με του php script στη λειτουργία List my no-activated slices με τη διαφορά ότι θέλουμε λίστα όλων των no-activated slices. Τα no-activated slices τα χωρίζουμε σε δύο καταστάσεις στις No-activated and No-reserved slices και No-activated and reserved slices, όπως αναφέραμε στο υποκεφάλαιο 5.1. Στην κατάσταση No-activated and No-reserved slices ανήκουν τα slices του χρήστη που ούτε είναι ενεργοποιημένα αυτή τη χρονική στιγμή αλλά ούτε έχουν δεσμευθεί για μελλοντική χρήση. Στην κατάσταση No-activated and reserved slices ανήκουν τα slices τα οποία δεν είναι ενεργοποιημένα αυτή τη χρονική στιγμή αλλά έχουν δεσμευθεί για μια ή περισσότερες χρονικές περιόδους στο μέλλον. Για να πάρουμε τις πληροφορίες για τα αντίστοιχα slices καλούμε τις απαραίτητες λειτουργίες από την υπηρεσία listsliceservice χωρίς κάποια παράμετρο και στη συνέχεια ακολουθούμε την ίδια διαδικασία με το script στο List my no-activated slices. Παραθέτουμε μια εικόνα της HTML σελίδας όπου μας δείχνει ότι συνολικά υπάρχουν πέντε no-activated slices τα οποία ανήκουν σε διαφορετικούς χρήστες και εκ των οποίων τα δύο ανήκουν στην υποκατηγορία no-reserved και τα τρία στην υποκατηγορία reserved:

List of no-activated Slices

List of no-reserved Slices which are not activate now

- The elena1 with owner elena contains the following nodes of testbed:
 1. The node3 with interfaces:
 - interface if0 with , Rate = 18 Mbps , RadioChannel = 3 , OperatingMode = STA
 2. The node4 with interfaces:
 - interface if1 with , Rate = 18 Mbps , RadioChannel = 1 , OperatingMode = AP
- The testelena6565 with owner elena contains the following nodes of testbed:
 1. The node3 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 2. The node5 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 3. The node7 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP

List of reserved Slices which are not activate now

- The testeur444 with owner evripidis has been reserved from 2011-07-01 16:00:00 to 2011-07-01 18:00:00. The slice contains the following nodes of testbed:
 1. The node8 with interfaces:
 - interface if0 with , Rate = 12 Mbps , RadioChannel = 1 , OperatingMode = AP
 2. The node10 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP
 - interface if1 with , Rate = 48 Mbps , RadioChannel = 11 , OperatingMode = STA
- The testeur7 with owner euri has been reserved from 2011-07-01 21:00:00 to 2011-07-01 23:00:00. The slice contains the following nodes of testbed:
 1. The node1 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 4 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 2 , OperatingMode = AP
 2. The node3 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 2 , OperatingMode = STA
 - interface if1 with , Rate = 12 Mbps , RadioChannel = 6 , OperatingMode = STA
- The testslice5 with owner evripidis has been reserved from 2011-07-01 18:00:00 to 2011-07-01 20:00:00. The slice contains the following nodes of testbed:
 1. The node1 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 2. The node3 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP

[Return to the services](#)

Σχήμα 6.26: Λίστα όλων των no-activated slices

List activated slices-Λίστα όλων των activated slices

Η λειτουργία αποτελείται από ένα php script στο οποίο ενσωματώνουμε μια HTML σελίδα και χρησιμοποιείται για να παρουσιάσει στο χρήστη σε μορφή λίστας το σύνολο των activated slices ανεξαρτήτως χρήστη. Το php script περιλαμβάνει διαδικασίες ίδιες με του php script στη λειτουργία List my activated slices με τη διαφορά ότι θέλουμε λίστα όλων των activated slices. Για αυτό το λόγο καλούμε τη λειτουργία listactivatedslices της υπηρεσίας listslicesservice χωρίς παράμετρο και στη συνέχεια ακολουθούμε την ίδια διαδικασία με το script στο List my activated slices. Παραθέτουμε μια εικόνα της HTML σελίδας όπου μας δείχνει ότι συνολικά υπάρχει ένα ενεργοποιημένο (activated) slice τη χρονική στιγμή που ζητήσαμε τη λίστα:

List of Activated Slices

- The testslice2 with owner evripidis ,which has been activated from 2011-07-01 14:00:00 and for 2 hours contains the following nodes of testbed:
 1. The node4 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = STA
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = AP
 2. The node5 with interfaces:
 - interface if0 with , Rate = 6 Mbps , RadioChannel = 3 , OperatingMode = AP
 - interface if1 with , Rate = 6 Mbps , RadioChannel = 1 , OperatingMode = STA

[Return to the services](#)

Σχήμα 6.27: Λίστα όλων των activated slices

Κεφάλαιο 7

Ανασκόπηση-Μελλοντική Εργασία

7.1 Ανασκόπηση

Στην παρούσα διπλωματική εργασία επεκτείνουμε μια υπάρχουσα δομή RSpec v.2 για την κατάλληλη περιγραφή ασύρματων πόρων. Συγκεκριμένα ορίσαμε τα απαραίτητα χαρακτηριστικά που πρέπει να έχουν οι ασύρματες διεπαφές των ασύρματων κόμβων. Τα χαρακτηριστικά αυτά είναι ο ρυθμός μετάδοσης, το κανάλι μετάδοσης και ο τρόπος λειτουργίας της εκάστοτε ασύρματης διεπαφής.

Στη συνέχεια βασιζόμενοι στη δομή δεδομένων RSpec v.2 και στην επέκταση που δημιουργήσαμε ώστε να περιγράψουμε κατάλληλα ασύρματους κόμβους σχεδιάσαμε ένα σύστημα που αναλύεται στο Κεφάλαιο 4 με το οποίο ο χρήστης θα μπορεί να δημιουργεί καινούργια slices και να διαχειρίζεται τα slices που έχει ήδη δημιουργήσει. Το σύστημα αυτό βασίζεται στον σχεδιασμό κατάλληλων υπηρεσιών διαδικτύου (web services) που επιτελούν τις διάφορες λειτουργίες διαχείρισης. Οι υπηρεσίες αυτές επικοινωνούν με μια βάση δεδομένων όπου υπάρχουν οι καταγραφές για όλα τα slices των χρηστών και παρέχουν τις απαιτούμενες πληροφορίες για τη διαχείρισή τους. Παράλληλα αναπτύξαμε ένα εργαλείο χρονοπρογραμματισμού σε κάποιες από τις υπηρεσίες διαδικτύου έτσι ώστε να δημοσιοποιούνται στο χρήστη μόνο οι διαθέσιμοι πόροι της πειραματικής πλατφόρμας όταν θέλει να δημιουργήσει κάποιο slice για κάποιο συγκεκριμένο χρονικό διάστημα. Αυτό έχει ως στόχο να μην υπάρχει χρονική επικάλυψη χρήσης των ίδιων πόρων της πειραματικής πλατφόρμας από δύο ή περισσότερα slices. Οι χρονικά διαμοιραζόμενοι πόροι της πειραματικής πλατφόρμας μεταξύ των διάφορων slices των ερευνητών είναι οι ασύρματοι κόμβοι και το σύνολο των καναλιών μετάδοσης που έχουμε ορίσει σε ένα slice.

7.2 Μελλοντική Εργασία

Η παρούσα διπλωματική εργασία δίνει τη δυνατότητα για αρκετές μελλοντικές επεκτάσεις τόσο στο κομμάτι της περιγραφής των ασύρματων πόρων όσο και στο κομμάτι της διαχείρισης των πόρων από τους ερευνητές.

Περιγραφή Ασύρματων Πόρων

Στο κομμάτι αυτό μπορούν να σχεδιαστούν κατάλληλες επεκτάσεις για την περιγραφή και άλλων χαρακτηριστικών των ασύρματων διεπαφών των ασύρματων κόμβων. Επιπλέον υπάρχει η δυνατότητα να γίνει κατάλληλη περιγραφή των ασύρματων ζεύξεων (link) με καθορισμό της χωρητικότητας (bandwidth) και του καναλιού μετάδοσης (radio channel) που χρησιμοποιεί η ζεύξη.

Διαχείριση ασύρματων πόρων

Στο κομμάτι αυτό μπορούν να γίνουν πολλές μελλοντικές επεκτάσεις και βελτιώσεις. Καταρχήν μπορεί να αναπτυχθεί κατάλληλος μηχανισμός για την επεξεργασία των απαιτούμενων χαρακτηριστικών που θέλει ο χρήστης για το πείραμα του και τον προσδιορισμό των κατάλληλων πόρων (resource mapping) που ικανοποιούν τις απαιτήσεις του χρήστη. Τέτοιοι μηχανισμοί έχουν αρχίσει να αναπτύσσονται για την διασύνδεση πειραματικών πλατφορμών ενσύρματων δικτύων στα πλαίσια του ευρωπαϊκού ερευνητικού προγράμματος FP7 NOVI [39]. Έτσι ο χρήστης θα μπορεί να δίνει μια περιγραφή των προδιαγραφών του πειράματος του και ο μηχανισμός αυτός θα εντοπίζει ποιοί είναι οι κατάλληλοι πόροι για το πείραμα του ερευνητή. Αυτό θα διευκολύνει το χρήστη να επιλέξει τους κατάλληλους πόρους για τα πειράματα και να τους δεσμεύσει για κάποιο δεδομένο χρονικό διάστημα εφόσον είναι διαθέσιμοι (resource allocation). Ένας παρόμοιος μηχανισμός για μέτρηση της ποιότητας των ζεύξεων μεταξύ των ασύρματων κόμβων με κάποια δεδομένα χαρακτηριστικά που έχει ορίσει ο χρήστης και προσδιορισμό των κατάλληλων πόρων με βάση τα αποτελέσματα των μετρήσεων υλοποιείται με βάση το πρωτόκολλο TLQAP (Topology and Link Quality Assessment Protocol) [40]. Στη συνέχεια ο μηχανισμός αυτός μπορεί να συνδεθεί με το χρονικό προγραμματιστή δέσμευσης πόρων (scheduler) ώστε οι κατάλληλοι πόροι που προκύπτουν από το resource mapping να δεσμεύονται (resource allocation) για κάποιο χρονικό διάστημα που είναι διαθέσιμοι.

Επιπλέον στην υλοποίηση του εργαλείου χρονοπρογραμματισμού (scheduler) μπορεί να υλοποιηθεί ένας αλγόριθμος για εντοπισμό του πόσο απομακρυσμένοι είναι οι κόμβοι που έχουν εκχωρηθεί στα διάφορα slices. Με τη βοήθεια αυτού του αλγόριθμου μπορούμε να κάνουμε πιο αποδοτικό χρονικό προγραμματισμό όσον αφορά τα κανάλια μετάδοσης που έχουν οριστεί στις ασύρματες διεπαφές των slices. Ο αλγόριθμος θα ανιχνεύει αν οι κόμβοι που επιλέγουμε για ένα slice είναι αρκετά απομακρυσμένοι από τους κόμβους των slices που είναι ενεργοποιημένα (activated) εκείνο το χρονικό διάστημα και θα επιτρέπει να ορίσουμε στις ασύρματες διεπαφές του καινούργιου slice τα κανάλια μετάδοσης που έχουμε σε αυτούς τους απομακρυσμένους κόμβους. Η υλοποίηση του αλγορίθμου θα βασίζεται στον καθορισμό των hops που πρέπει να απέχουν οι κόμβοι για να μην υπάρχει παρεμβολή μεταξύ τους και επομένως να μπορούμε να χρησιμοποιήσουμε το ίδιο κανάλι στις διεπαφές τους. Με αυτό τον τρόπο θα πετύχουμε πιο αποδοτική χρήση των πόρων της πειραματικής πλατφόρμας κυρίως όταν αυτή περιέχει μεγάλο αριθμό κόμβων.

Παράρτημα: Βιβλιογραφία

- [1] Yuen, M. (2010), GENI in the Cloud, A Dissertation submitted in partial fulfillment of the Requirements for the Degree of MASTER OF SCIENCE in the Department of Computer Science, University of Victoria, 2010
- [2] RSpec-ProtoGENI, <http://www.protogeni.net/trac/protogeni/wiki/RSpec>
- [3] Angelos-Christos Anadiotis, Apostolos Apostolaras, Dimitris Syrivelis, Thanasis Korakis, Leandros Tassioulas, Luis Rodriguez, Ivan Seskar, and Maximilian Ott, "Towards Maximizing Wireless Testbed Utilization using Spectrum Slicing", Proceedings of TridentCom 2010. Berlin (Germany): May 2010.
- [4] Thierry Rakotoarivelo, Maximilian Ott, Guillaume Jourjon, Ivan Seskar, "OMF: A Control and Management Framework for Networking Testbeds", ACM SIGOPS Operating Systems Volume 43 Issue 4, January 2010, pages 54-59
- [5] PlanetLab: An Open Platform for Developing, Deploying and Accessing Planetary-scale services, <http://www.planet-lab.org>
- [6] PlanetLab Implementation of the Slice-Based Facility Architecture (SFA), version 0.05, June 23, 2009, Ανακτήθηκε από <http://www.cs.princeton.edu/~llp/geniwrapper.pdf>
- [7] KanseiGenieSensorNet, <http://www.groups.geni.net/geni/wiki/KanseiSensorNet>
- [8] Xi Ju, Hongwei Zhang, Wenjie Zeng, Mukundan Sridharan, Jing Li, Anish Arora, Rajiv Ramnath, Yufeng Xin, LENS: Resource Specification for Wireless Sensor Network Experimentation Infrastructures, *6th ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH)* (in conjunction with ACM MobiCom'11), 2011
- [9] GENI - ORBIT Management Framework (OMF), <http://www.winlab.rutgers.edu/docs/focus/GENI-OMF.html>
- [10] Network Description Language, <http://www.science.uva.nl/research/sne/ndl>
- [11] Network Description Language plus OWL, <http://geni-orca.renci.org/trac/wiki/NDL-OWL>
- [12] Network markup Language, <http://forge.ogf.org/sf/projects/nml-wg>
- [13] NITOS Wireless Testbed – Network Implementation Testbed Laboratory, <http://nitlab.inf.uth.gr/NITlab/index.php/testbed>

- [14] Anish Arora, Hongwei Zhang, Rajiv Ramnath, Mukundan Sridharan, Wenjie Zeng, Xi Ju, William Leal, "*KanseiGenie: Software infrastructure for resource management and programmability of wireless sensor network fabrics*", in Next-Generation Internet: Architectures and Protocols, pages 275-298, Cambridge University Press 2011
- [15] Extensible Markup Language (XML), <http://www.w3.org/XML>
- [16] Wikipedia the free encyclopedia, <http://www.wikipedia.org>
- [17] HyperText Markup Language (HTML), <http://www.w3.org/TR/html5/>
- [18] W3C XML Schema, <http://www.w3.org/XML/Schema>
- [19] Namespaces in XML 1.0 (Third Edition), <http://www.w3.org/TR/xml-names/>
- [20] RELAX NG Compact Syntax, <http://relaxng.org/compact-tutorial-20030326.html>
- [21] W3Schools Online Web Tutorials, <http://www.w3schools.com>
- [22] Namespace-based Validation Dispatching Language, <http://www.nvdl.org>
- [23] NVDL Tutorial, <http://jnvdl.sourceforge.net/tutorial.html>
- [24] ProtoGENI-URNs, <http://www.protogeni.net/trac/protogeni/wiki/URNs>
- [25] Global Environment for Network Innovations (GENI), <http://www.geni.net>
- [26] Web Services Architecture, <http://www.w3.org/TR/ws-arch/>
- [27] Simple Object Access Protocol (SOAP), <http://www.w3.org/TR/soap>
- [28] Web Service Definition Language (WSDL), <http://www.w3.org/TR/wsdl>
- [29] WSO2 Web Services Framework for PHP, <http://wso2.com/products/web-services-framework/php/>
- [30] WSO2 WSF/PHP Manual, <http://wso2.org/project/wsf/php/2.0.0/docs/manual.html>
- [31] J.F.Kurose, K.W. Ross, "*Δικτύωση Υπολογιστών*", σελίδες 526-527 , 4^η Έκδοση, Εκδόσεις Μ. Γκίουρδας, 2008
- [32] JNVDL - an Open-source Java-based Implementation of NVDL, <http://jnvdl.sourceforge.net/>
- [33] HyperText Transfer Protocol (HTTP), <http://www.w3.org/Protocols/>
- [34] MySQL, The world's most popular open source database, <http://www.mysql.com>
- [35] T. Ylonen and C. Lonvick, "*The Secure Shell (SSH) Protocol Architecture, RFC 4251 (Proposed Standard)*", January 2006
- [36] PHP: Web Services-Manual, <http://php.net/manual/en/refs.webservice.php>
- [37] SQL Tutorial – Learn SQL, <http://www.sql-tutorial.net>

- [38] Forms in HTML Document, <http://www.w3.org/TR/html4/interact/forms.html>
- [39] NOVI (Network innovations Over Virtualized Infrastructures), <http://www.fp7-novi.eu>
- [40] D. Syrivelis, A. C. Anadiotis, A. Apostolaras, T. Korakis, L. Tassiulas. TLQAP: A topology and link quality assessment protocol for efficient node allocation on wireless testbeds. In proceedings of ACM WiNTECH. Beijing, China September 21, 2009.

Παράρτημα Β: Κώδικας

loginservice.php

```
<?php

function login($message){

    $simplexml = new SimpleXMLElement($message->str);

    $username =$simplexml->username;
    $password=$simplexml->password;

    $host = "localhost";
    $user = "omf";
    $pass = "omf";
    $database = "inventory2";

    $linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
    mysql_select_db($database, $linkID) or die("Could not find database.");

    $query1="SELECT * FROM login WHERE username='$username' AND password='$password'";

    $result = mysql_query($query1, $linkID) or die("Data not found.");

    $row = mysql_fetch_assoc($result);
    if ($row>=1)
    {
        $responsePayloadString = <<<XML
        <loginresponse>
        <result>go</result>
        </loginresponse>
        XML;

        $outMessage = new WSMMessage($responsePayloadString);

        return $outMessage;
    }
    else
    {
        $responsePayloadString = <<<XML
        <loginresponse>
        <result>wrong</result>
        </loginresponse>
        XML;

        $outMessage = new WSMMessage($responsePayloadString);

        return $outMessage;
    }
}

$operations = array("login"=>"login");

$service = new WSService(array ("operations" => $operations));
// process client requests and reply

$service->reply();
?>
```

login.php

```
<?php
//html page for the user authentication
Echo "<html>";
Echo "<title>Login to use the services</title>";

Echo "<h1>Give the username and the password</h1>";
```

```

Echo "<form method=\"post\" action=\"http://sorch.netmode.ntua.gr/ws/login2.php\">";

?>
Username:<input type="text" size="20" maxlength="20" name="username" ><br />
Password:<input type="password" size="20" maxlength="20" name="password" ><br />

<input type="submit" value="submit" name="login">
<input type="reset" value="Reset" name="reset">

</form>

</html>

```

login2.php

```

<?php
//php script for checking username/password
$login = isset($_POST["login"]) ? $_POST['login'] : '';
$username = isset($_POST["username"]) ? $_POST['username'] : '';
$password = isset($_POST["password"]) ? $_POST['password'] : '';

if (isset($_POST['login']))
{
    if (isset($_GET['username']) && isset($_GET['password']))
    {
        $username = $_GET['username'];
        $password=$_GET['password'];
    }

    if (isset($username) && isset($password) && ($username<>'') && ($password<>''))
    {
        try {

            $client = new WSClient(array("to" => "http://sorch.netmode.ntua.gr/ws/loginservice.php", "useSOAP"
=> 1.1));

            $reqPayloadString1 = <<<XML
<ns1:login xmlns:ns1="http://sorch.netmode.ntua.gr/ws/loginservice.php">
<username>$username</username><password>$password</password>
</ns1:login>
XML;

            $reqMessage1 = new WSMMessage($reqPayloadString1);

            $response1 = $client->request($reqMessage1);
            $simplexmll1 = new SimpleXMLElement($response1->str);

            $result=$simplexmll1->result;

            if (strcmp($result,"go")==0)
            {
                session_start();

                $_SESSION['username'] = $username;
                $_SESSION['login'] ="go";
                header("Location:services.php");
            }
            else
            {
                echo "Wrong username or password.<br/>";
                echo "<a href=\"http://sorch.netmode.ntua.gr/ws/login.php\">Please try again</a><br/>";
            }
        } catch (Exception $e) {

            if ($e instanceof WSFault) {
                printf("Soap Fault: %s\n", $e->Reason);
            } else {
                printf("Message = %s\n",$e->getMessage());
            }
        }
    }
    else
    {
        echo "You have not put all the parameters needed.<a
href=\"http://sorch.netmode.ntua.gr/ws/login.php\">Please try again</a><br/>";
    }
}

```

```
}  
}  
>
```

logout.php

```
<?php  
//logout php script  
session_start();  
session_destroy();  
header("Location:login.php");  
>
```

mksliceservice.php

```
<?php  
  
class RSpec {  
public $name;  
public $owner;  
public $timestamp;  
public $duration;  
public $rspec1;  
}  
  
//function mkslice for the creation and reservation of a slice based on the available nodes in  
specific time  
function mkslice($message){  
  
$host = "localhost";  
$user = "omf";  
$pass = "omf";  
$database = "inventory2";  
  
$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");  
mysql_select_db($database, $linkID) or die("Could not find database.");  
  
$name=$message->name;  
$owner=$message->owner;  
$timestamp=$message->timestamp;  
$duration=$message->duration;  
$rspec=$message->rspec1;  
  
$simplexml = new SimpleXMLElement($rspec);  
$xdoc = new DomDocument();  
$xmlschema = '/var/www/ws/RSpec/request.xsd';  
//Load the xml document in the DOMDocument object  
$xdoc->loadXML($rspec);  
  
if ($xdoc->schemaValidate($xmlschema)) {  
  
$query1="SELECT * FROM slices WHERE name='$name'";  
  
$result = mysql_query($query1, $linkID) or die("Data not found.");  
  
$row = mysql_fetch_assoc($result);  
if ($row>=1)  
{  
$response="The slice with name ".$name." already exists so try making another slice";  
return $response;  
}  
else  
{  
  
$query="INSERT INTO slices (`slice_id`,`name`,`owner`,`number_nodes`) VALUES  
(','.$name','.$owner','0)";  
  
mysql_query($query, $linkID);  
  
$query2="SELECT * FROM slices WHERE name='$name'";  
  
$result2 = mysql_query($query2, $linkID) or die("Data not found.");  
  
$row2 = mysql_fetch_assoc($result2);
```

```

$sliceid=$row2['slice_id'];

$response="You made the slice ".$name."<br/>";
if (!$simplexml->node)
{
$response.="You have not added any resources to the slice ".$name."<br/>";
}
else
{
foreach ($simplexml->node as $entry2){
$nodename=$entry2['client_id'];

$querycheck="SELECT * FROM nodes WHERE hrn='$nodename'";

$resultcheck = mysql_query($querycheck, $linkID) or die("Data not found.");

$rowcheck = mysql_fetch_assoc($resultcheck);

$hrn=$rowcheck['hrn'];
$id=$rowcheck['id'];
$hostname1=$rowcheck['hostname'];

$query1="INSERT INTO slice_node_association (`slice_id`,`node_id`) VALUES ('$sliceid','$id')";

mysql_query($query1,$linkID);

$query1="SELECT number_nodes FROM slices WHERE name='$name'";
$result1 = mysql_query($query1, $linkID) or die("Data not found.");

$row1 = mysql_fetch_assoc($result1);
$number=$row1['number_nodes'];

$numbernew=$number+1;

$query3="UPDATE slices SET number_nodes='$numbernew' WHERE name='$name'";

mysql_query($query3,$linkID);

$response.="You added the ".$hostname1." to the slice ".$name;

if (!$entry2->interface)
{
$response.=" with no interfaces <br/>";
}
else
{
$response.=" with interfaces <br/><ul>";
foreach ($entry2->interface as $entry){

$check1=$entry['client_id'];
$array=split(":",$check1);
$x=$array[1];

$dc = $entry->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc->Wireless_features as $entry1){
$response.="<li> interface ".$x." with ";

$query3="INSERT INTO node_if_association (`slice_id`,`node_id`,`if_id`)
VALUES ('$sliceid','$id','$x')";
mysql_query($query3, $linkID);

$dc1 = $entry1->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){

$nodeName=$features->getName();

$value=$entry1->$nodeName;
$response.=" ".$nodeName." = ".$value;

$query4="UPDATE node_if_association SET $nodeName='$value' WHERE slice_id='$sliceid' AND
node_id='$id' AND if_id='$x'";
mysql_query($query4, $linkID);

```

```

}
$response.=" </li>";

}

}
$response.="</ul>";
}
}

$querytimestamp="SELECT UNIX_TIMESTAMP('$timestamp')";
$resulttimestamp=mysql_query($querytimestamp, $linkID);
$rowtimestamp = mysql_fetch_assoc($resulttimestamp);
$timestamp_start=$rowtimestamp["UNIX_TIMESTAMP('$timestamp')"];

$time=time();

$duration_timestamp=3600*$duration;

$timestamp_end=$timestamp_start+$duration_timestamp;

$querytimestamp_final="SELECT FROM_UNIXTIME('$time')";
$resulttimestamp_final=mysql_query($querytimestamp_final, $linkID);
$rowtimestamp_final = mysql_fetch_assoc($resulttimestamp_final);
$timestamp_start1=$rowtimestamp_final["FROM_UNIXTIME('$time')"];

$querytimestamp_final2="SELECT FROM_UNIXTIME('$timestamp_end')";
$resulttimestamp_final2=mysql_query($querytimestamp_final2, $linkID);
$rowtimestamp_final2 = mysql_fetch_assoc($resulttimestamp_final2);
$timestamp_end_final=$rowtimestamp_final2["FROM_UNIXTIME('$timestamp_end')"];

if ($timestamp_start<=$time)
{
$query1="UPDATE slices SET activate=1,duration_activation='$duration',timeactivation=NOW() WHERE
name='$name'";
mysql_query($query1, $linkID);

$query3="INSERT INTO inventory2.reservations
(`name`,`time_activation_start`,`time_activation_end`)
VALUES('$name','$timestamp_start1','$timestamp_end_final')";
mysql_query($query3, $linkID);

$event=$name."_".$time;
$query2="CREATE EVENT $event
ON SCHEDULE AT '$timestamp_start1' + INTERVAL '$duration' HOUR
DO
UPDATE inventory2.slices SET activate=0,duration_activation='0',timeactivation='0000-00-
00 00:00:00' WHERE name='$name'";

mysql_query($query2, $linkID);

$response.="<br/>".$owner." has reserved the slice ".$name." at ".$timestamp_start1." for
".$duration." hours.";

}
else
{
$event1=$name."_2".$timestamp_start;
$query1="CREATE EVENT $event1
ON SCHEDULE AT '$timestamp'
DO
UPDATE inventory2.slices SET
activate=1,duration_activation='$duration',timeactivation='$timestamp' WHERE name='$name'";

mysql_query($query1, $linkID);
$query3="INSERT INTO inventory2.reservations
(`name`,`time_activation_start`,`time_activation_end`)
VALUES('$name','$timestamp','$timestamp_end_final')";
mysql_query($query3, $linkID);

$event=$name."_".$timestamp_start;

```



```

$query2="CREATE EVENT $event
        ON SCHEDULE AT '$timestamp' + INTERVAL '$duration' HOUR
        DO
            UPDATE inventory2.slices SET activate=0,duration_activation='0',timeactivation='0000-00-
00 00:00:00' WHERE name='$name';

mysql_query($query2, $linkID);

$response="<br/>".$owner." has reserved the slice ".$name." at ".$timestamp." for ".$duration."
hours.";
}

return $response;
}
}
else
{
return "not valid";
}
}

//function modifyslice for the modification of a specific slice using the available //resources
function modifyslice($message){

$host = "localhost";
$user = "omf";
$pass = "omf";
$databse = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($databse, $linkID) or die("Could not find databse.");

$name=$message->name;
$rspec=$message->rspec1;

$simplexml = new SimpleXMLElement($rspec);

$xml = new DomDocument();
$xmlschema = '/var/www/ws/RSpec/request.xsd';
//Load the xml document in the DOMDocument object
$xml->loadXML($rspec);

if ($xml->schemaValidate($xmlschema)) {

$query1="SELECT * FROM slices WHERE name='$name';

$result = mysql_query($query1, $linkID) or die("Data not found.");

$row = mysql_fetch_assoc($result);

$owner=$row['owner'];
$slice_id=$row['slice_id'];

$query1="UPDATE slices SET number_nodes=0 WHERE name='$name';
mysql_query($query1, $linkID);

$query2="DELETE FROM slice_node_association WHERE slice_id='$slice_id';
mysql_query($query2, $linkID);

$query4="DELETE FROM node_if_association WHERE slice_id='$slice_id';
mysql_query($query4, $linkID);

$query2="SELECT * FROM slices WHERE name='$name';

$result2 = mysql_query($query2, $linkID) or die("Data not found.");

$row2 = mysql_fetch_assoc($result2);
$sliceid=$row2['slice_id'];

$response="You modified the slice ".$name."<br/>";

```

```

if (!$simplexml->node)
{
$response.="You have not added any resources to the slice ".$name."<br/>";
}
else
{
foreach ($simplexml->node as $entry2){
$check=array();
$message="";
$nodename=$entry2['client_id'];

$querycheck="SELECT * FROM nodes WHERE hrn='$nodename'";

$resultcheck = mysql_query($querycheck, $linkID) or die("Data not found.");

$rowcheck = mysql_fetch_assoc($resultcheck);

$id=$rowcheck['id'];

$query_check1="SELECT * FROM reservations WHERE name='$name'";

$result_check1 = mysql_query($query_check1, $linkID) or die("Data not found.");
$num_rows=mysql_num_rows($result_check1);

if($num_rows<1)
{
$check[]=1;
}
else
{
for($x = 0 ; $x < mysql_num_rows($result_check1) ; $x++){
$row_check1 = mysql_fetch_assoc($result_check1);

$timestamp_start_check=$row_check1['time_activation_start'];
$timestamp_end_check=$row_check1['time_activation_end'];

$querytimestamp1="SELECT UNIX_TIMESTAMP('$timestamp_start_check')";
$resulttimestamp1=mysql_query($querytimestamp1, $linkID);
$rowtimestamp1 = mysql_fetch_assoc($resulttimestamp1);
$timestamp_start1=$rowtimestamp1["UNIX_TIMESTAMP('$timestamp_start_check')"];

$querytimestamp2="SELECT UNIX_TIMESTAMP('$timestamp_end_check')";
$resulttimestamp2=mysql_query($querytimestamp2, $linkID);
$rowtimestamp2= mysql_fetch_assoc($resulttimestamp2);
$timestamp_end1=$rowtimestamp2["UNIX_TIMESTAMP('$timestamp_end_check')"];

$time=time();

if (!(($timestamp_end1<=$time))
{
$query_check6="SELECT * FROM reservations WHERE name<>'$name'";
$result_check6 = mysql_query($query_check6, $linkID) or die("Data not found.");
$num_rows4=mysql_num_rows($result_check6);
if ($num_rows4<1){
$check[]=1;
}
else {
for($z = 0 ; $z < mysql_num_rows($result_check6) ; $z++){
$row_check6 = mysql_fetch_assoc($result_check6);

$timestamp_start_check2=$row_check6['time_activation_start'];
$timestamp_end_check2=$row_check6['time_activation_end'];

$querytimestamp3="SELECT UNIX_TIMESTAMP('$timestamp_start_check2')";
$resulttimestamp3=mysql_query($querytimestamp3, $linkID);
$rowtimestamp3 = mysql_fetch_assoc($resulttimestamp3);
$timestamp_start2=$rowtimestamp3["UNIX_TIMESTAMP('$timestamp_start_check2')"];

$querytimestamp4="SELECT UNIX_TIMESTAMP('$timestamp_end_check2')";
$resulttimestamp4=mysql_query($querytimestamp4, $linkID);
$rowtimestamp4= mysql_fetch_assoc($resulttimestamp4);
$timestamp_end2=$rowtimestamp4["UNIX_TIMESTAMP('$timestamp_end_check2')"];

$check6=($timestamp_start1<=$timestamp_start2) && ($timestamp_end1<=$timestamp_start2);
$check7=($timestamp_start1>=$timestamp_end2) && ($timestamp_end1>=$timestamp_end2);

```

```

if ($check6 || $check7)
{
$check[]=1;
}
else
{
$name_check=$row_check6['name'];
$query_check5="SELECT * FROM slices WHERE name='$name_check'";
$result_check5 = mysql_query($query_check5, $linkID) or die("Data not found.");
$row_check5= mysql_fetch_assoc($result_check5);
$slice_id_check=$row_check5['slice_id'];
$owner=$row_check5['owner'];

$query_check4="SELECT * FROM slice_node_association WHERE slice_id={$slice_id_check} AND
node_id={$id}";
$result_check4 = mysql_query($query_check4, $linkID) or die("Data not found.");
$num_rows3=mysql_num_rows($result_check4);

if ($num_rows3<1)
{
$check[]=1;
}
else
{
$check[]=2;

$message.="Node ".$id." is reserved by slice ".$name_check." with owner ".$owner." from
".$timestamp_start_check2." to ".$timestamp_end_check2." <br/>";
}
}
}
}
}

if (!in_array(2,$check)){

$hrn=$rowcheck['hrn'];
$hostname1=$rowcheck['hostname'];
$query1="INSERT INTO slice_node_association (`slice_id`,`node_id`) VALUES ('$sliceid','$id')";

mysql_query($query1,$linkID);

$query1="SELECT number_nodes FROM slices WHERE name='$name'";
$result1 = mysql_query($query1, $linkID) or die("Data not found.");

$row1 = mysql_fetch_assoc($result1);
$number=$row1['number_nodes'];

$numbernew=$number+1;

$query3="UPDATE slices SET number_nodes='$numbernew' WHERE name='$name'";

mysql_query($query3,$linkID);
$response.="The ".$hostname1." to the slice ".$name;
if (!$entry2->interface)
{
$response.=" with no interfaces <br/>";
}
else
{
$response.=" with interfaces <br/><ul>";
foreach ($entry2->interface as $entry){

$check1=$entry['client_id'];
$array=split(":",$check1);
$x=$array[1];

$dc = $entry->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc->Wireless_features as $entry1){
$response.="<li> interface ".$x." with ";

```

```

$query3="INSERT INTO node_if_association (`slice_id`,`node_id`,`if_id`)
VALUES('$sliceid','$id','$x');
mysql_query($query3, $linkID);

$dc1 = $entry1->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){

$nodeName=$features->getName();

$value=$entry1->$nodeName;
$response.=" ".$nodeName." = ".$value;

$query4="UPDATE node_if_association SET $nodeName='$value' WHERE slice_id='$sliceid' AND
node_id='$id' AND if_id='$x';
mysql_query($query4, $linkID);
}
$response.=" </li>";
}
}
$response.="</ul>";
}
}
else
{
$response.=$message;
}
}
}

return $response;
}
else
{
return "not valid";
}
}

ini_set('soap.wsdl_cache_enabled', '0');
$server = new SoapServer("sliceoperations.wsdl");
$server->addFunction("mkslice");
$server->addFunction("modifyslice");
$server->handle();
?>

```

sliceoperations.wsdl (wsdl αρχείο για την περιγραφή της παραπάνω υπηρεσίας)

```

<?xml version='1.0' encoding='UTF-8' ?>
<definitions name='RSpecSlice'
targetNamespace='http://sorch.netmode.ntua.gr/ws'
xmlns:tns='http://sorch.netmode.ntua.gr/ws'
xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'
xmlns:xsd1='http://sorch.netmode.ntua.gr/ws/'
xmlns='http://schemas.xmlsoap.org/wsdl/'>

<types>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://sorch.netmode.ntua.gr/ws"
targetNamespace="http://sorch.netmode.ntua.gr/ws"
xmlns:xsd1="http://sorch.netmode.ntua.gr/ws/">

<complexType name="RSpec">
<sequence>
<element name="name" type="xsd:string"/>
<element name="owner" type="xsd:string"/>
<element name="timestamp" type="xsd:string"/>
<element name="duration" type="xsd:int"/>
<element name="rspec1" type="xsd:string"/>
</sequence>
</complexType>
</schema>
</types>

<message name='mksliceRequest'>
<part name='Result' type='xsd1:RSpec'/>
</message>

```

```

<message name='mksliceResponse'>
  <part name='Result' type='xsd:string' />
</message>

<message name='modifysliceRequest'>
  <part name='Result' type='xsd1:RSpec' />
</message>

<message name='modifysliceResponse'>
  <part name='Result' type='xsd:string' />
</message>

<portType name='sliceoperationsPortType'>
  <operation name='mkslice'>
    <input message='tns:mksliceRequest' />
    <output message='tns:mksliceResponse' />
  </operation>

  <operation name='modifyslice'>
    <input message='tns:modifysliceRequest' />
    <output message='tns:modifysliceResponse' />
  </operation>
</portType>

<binding name='sliceoperationsBinding' type='tns:sliceoperationsPortType'>
<soap:binding style='rpc' transport='http://schemas.xmlsoap.org/soap/http' />

<operation name='mkslice'>
<soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/mkslice' />
<input>
<soap:body use='encoded' namespace='http://sorch.netmode.ntua.gr/ws'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding' />
</input>
<output>
<soap:body use='encoded' namespace='http://sorch.netmode.ntua.gr/ws'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding' />
</output>
</operation>

<operation name='modifyslice'>
<soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/modifyslice' />
<input>
<soap:body use='encoded' namespace='http://sorch.netmode.ntua.gr/ws'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding' />
</input>
<output>
<soap:body use='encoded' namespace='http://sorch.netmode.ntua.gr/ws'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding' />
</output>
</operation>

</binding>

<service name='mksliceservice'>
<port name='sliceoperationsPortType' binding='sliceoperationsBinding'>
  <soap:address location='http://sorch.netmode.ntua.gr/ws/mksliceservice.php' />
</port>
</service>

</definitions>

```

check_time.php

```

<?php
//php scripts in order to give the time you want to reserve the slice

Echo "<html>";
Echo "<title>Give the time</title>";

Echo "<h1>Give the time when you want to use the slice to see what resources are
available:</h1>";

Echo "<form method='post' action = 'http://sorch.netmode.ntua.gr/ws/mkslicenew2.php'>";
?>

Put the date that you want to activate the slice:

<select name="month">
  <option> - Month - </option>
  <option value="01">January</option>
  <option value="02">February</option>
  <option value="03">March</option>
  <option value="04">April</option>
  <option value="05">May</option>
  <option value="06">June</option>
  <option value="07">July</option>

```

```

    <option value="08">August</option>
    <option value="09">September</option>
    <option value="10">October</option>
    <option value="11">November</option>
    <option value="12">December</option>
</select>

```

```

/<select name="day">
  <option> - Day - </option>
  <option value="01">1</option>
  <option value="02">2</option>
  <option value="03">3</option>
  <option value="04">4</option>
  <option value="05">5</option>
  <option value="06">6</option>
  <option value="07">7</option>
  <option value="08">8</option>
  <option value="09">9</option>
  <option value="10">10</option>
  <option value="11">11</option>
  <option value="12">12</option>
  <option value="13">13</option>
  <option value="14">14</option>
  <option value="15">15</option>
  <option value="16">16</option>
  <option value="17">17</option>
  <option value="18">18</option>
  <option value="19">19</option>
  <option value="20">20</option>
  <option value="21">21</option>
  <option value="22">22</option>
  <option value="23">23</option>
  <option value="24">24</option>
  <option value="25">25</option>
  <option value="26">26</option>
  <option value="27">27</option>
  <option value="28">28</option>
  <option value="29">29</option>
  <option value="30">30</option>
  <option value="31">31</option>
</select>

```

```

/<select name="year">
  <option> - Year - </option>
  <option value="2011">2011</option>
  <option value="2012">2012</option>
  <option value="2013">2013</option>
  <option value="2014">2014</option>
  <option value="2015">2015</option>
  <option value="2016">2016</option>
  <option value="2017">2017</option>
  <option value="2018">2018</option>
  <option value="2019">2019</option>
  <option value="2020">2020</option>
</select>
<br/>

```

Put the hour that you want to activate the slice:

```

<select name="hour_start">
  <option> - Start Hour - </option>
  <option value="00">0 am</option>
  <option value="01">1 am</option>
  <option value="02">2 am</option>
  <option value="03">3 am</option>
  <option value="04">4 am</option>
  <option value="05">5 am</option>
  <option value="06">6 am</option>
  <option value="07">7 am</option>
  <option value="08">8 am</option>
  <option value="09">9 am</option>
  <option value="10">10 am</option>
  <option value="11">11 am</option>
  <option value="12">12 pm</option>
  <option value="13">1 pm</option>
  <option value="14">2 pm</option>
  <option value="15">3 pm</option>
  <option value="16">4 pm</option>
  <option value="17">5 pm</option>
  <option value="18">6 pm</option>
  <option value="19">7 pm</option>
  <option value="20">8 pm</option>
  <option value="21">9 pm</option>
  <option value="22">10 pm</option>
  <option value="23">11 pm</option>
</select>
<br />

```

```

Put the time that you want to use the slice in hours:<input type="text" size="2" maxlength="2"
name="hour">
<br />

<input type="submit" value="submit" name="getSlice">
<input type="reset" value="Reset" name="reset">

</form>

</html>

```

mkslicenew2.php

```

<?php
//php script for choosing which resources from the available ones you want to allocate //in a
slice

$getSlice = isset($_POST["getSlice"]) ? $_POST['getSlice'] : '';
$hour=isset($_POST["hour"]) ? $_POST['hour'] : '';
$month=isset($_POST["month"]) ? $_POST['month'] : '';
$year=isset($_POST["year"]) ? $_POST['year'] : '';
$day=isset($_POST["day"]) ? $_POST['day'] : '';
$hour_start=isset($_POST["hour_start"]) ? $_POST['hour_start'] : '';

Echo "<html>";
Echo "<title>Make a new slice</title>";

if (isset($_POST['getSlice']))
{
    if ( isset($_GET['hour']) && isset($_GET['hour_start']) && isset($_GET['month']) && isset
($_GET['day']) && isset($_GET['year']))
    {
        $hour=$_GET['hour'];
        $hour_start=$_GET['hour_start'];
        $day=$_GET['day'];
        $month=$_GET['month'];
        $year=$_GET['year'];
    }

$date=$year."-".$month."-".$day;

    $hour_init=$hour_start.".00:00";

$timestamp=$date." ".$hour_init;

$timestamp_check=strtotime($timestamp);
$time_now=time();

if (!(checkdate($month,$day,$year)) || ($timestamp_check<$time_now))
{
    Echo "You have not put right the time.<a
href=\"http://sorch.netmode.ntua.gr/ws/check_time.php\">Please try again</a><br/><br/>";
}
else
{
    $channels=array("IEEE 802.11b/g:Channel 1:2412MHz","IEEE 802.11b/g:Channel 2:2417MHz","IEEE
802.11b/g:Channel 3:2422MHz","IEEE 802.11b/g:Channel 4:2427MHz","IEEE 802.11b/g:Channel
5:2432MHz","IEEE 802.11b/g:Channel 6:2437MHz","IEEE 802.11b/g:Channel 7:2442MHz","IEEE
802.11b/g:Channel 8:2447MHz","IEEE 802.11b/g:Channel 9:2452MHz","IEEE 802.11b/g:Channel
10:2457MHz","IEEE 802.11b/g:Channel 11:2462MHz","IEEE 802.11b/g:Channel 12:2467MHz","IEEE
802.11b/g:Channel 13:2472MHz");

$time=$year." ".$month." ".$day." ".$hour_start." ".$hour;

$client = new SoapClient("listslices12.wsdl");
$response1 = $client->getAvailableResources($time);

$response2=array();
$nodes=array();
$interfacesarray=array();

if (strcmp($response1,"not valid")==0)
{
    echo "not valid";
}
else
{
    Echo "<h1>Give the name of slice and select the nodes that you prefer for the new slice</h1>";

Echo "<form method=\"post\" action=\"http://sorch.netmode.ntua.gr/ws/mkslice2.php\">";

?>

Name of Slice:<input type="text" size="60" maxlength="60" name="name" ><br />


```

```

Choose your nodes for the slice :<br><br>

<?php
$i=0;
$x=0;

$simplexml = new SimpleXMLElement($response1);

foreach ($simplexml->node as $entry4){
$nodes[]=$entry4;
}

$node=$nodes[$x];

foreach ($node->interface as $interfacel)
{
$interfacesarray[]=$interfacel;
}
$interface2=$interfacesarray[$x];

$dc = $interface2->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc->Wireless_features as $entry1){
$dcl = $entry1->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dcl->RadioChannel as $entry3){
$response2[]=$entry3;
}
foreach ($dcl->Rate as $entry3){
$rates=$entry3;
}
foreach ($dcl->OperatingMode as $entry3){
$mode[]=$entry3;
}
}

$rates1=split(" ",$rates);
for ($i=0;$i<count($rates1);$i++)
{
$rate[]=$rates1[$i];
}

foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\+",$string1);
$hostname=$array1[3];

$id=preg_replace("/^[^0-9s]/", "", $hostname);

Echo"<b><input type=\"checkbox\" name=\"nodes[]\" value=$id> $hostname <br></b>";

foreach ($entry2->interface as $interfacel)
{
$component_id2=$interfacel['component_name'];
$array=split(':', $component_id2);
$if=$id.".".$array[1].".".$x;
Echo" <input type=\"checkbox\" name=\"interfaces[]\" value=$if> $hostname $array[1] <br>";
?>

<select name="rate[]">
<?php
for ($q=0;$q<count($rate);$q++){
$temp=$rate[$q];
?>
<option value="<?php echo $temp ?>"><?php echo $temp ?> Mbps</option>
<?php
}
?>
</select>

<select name="channel[]">
<?php
for ($q=0;$q<count($response2);$q++){
$temp=$response2[$q];
?>
<option value="<?php echo $temp ?>"><?php echo $channels[$temp-1] ?> </option>
<?php
}
?>

```



```

</select>

<select name="mode[]">
<?php
for ($q=0;$q<count($mode);$q++){
$temp=$mode[$q];
?>
<option value="<?php echo $temp ?>"><?php echo $temp ?> </option>
<?php
}
?>
</select>
<br/>
<?php
$i=$i+1;
}
Echo"<br/>";

$x=$x+1;
}

Echo "<input type=\"hidden\" name=\"date\" value=$date />";
Echo "<input type=\"hidden\" name=\"hour_init\" value=$hour_init />";
Echo "<input type=\"hidden\" name=\"duration\" value=$hour />";

?>

<br/>
<input type="submit" value="submit" name="getSlice">
<input type="reset" value="Reset" name="reset">
<?php
}
}
}
?>
</form>

</html>

```

mkslice2.php

```

<?php
//php script for calling the operation mkslice in order to create a new slice with //specific
resources
$getSlice = isset($_POST["getSlice"]) ? $_POST['getSlice'] : '';
$nodes = isset($_POST["nodes"]) ? $_POST['nodes'] : '';
$interfaces= isset($_POST["interfaces"]) ? $_POST['interfaces'] : '';
$rate=isset($_POST["rate"]) ? $_POST['rate'] : '';
$mode=isset($_POST["mode"]) ? $_POST['mode'] : '';
$channel=isset($_POST["channel"]) ? $_POST['channel'] : '';
$name_init=isset($_POST["name"]) ? $_POST['name'] : '';
//$owner=isset($_POST["owner"]) ? $_POST['owner'] : '';
$date=isset($_POST["date"]) ? $_POST['date'] : '';
$hour_init=isset($_POST["hour_init"]) ? $_POST['hour_init'] : '';
$duration=isset($_POST["duration"]) ? $_POST['duration'] : '';
session_start();
$owner=$_SESSION['username'];

class RSpec {
public $name;
public $owner;
public $timestamp;
public $duration;
public $rspec1;
}

?>

<html>
<head>
<title>Make a slice</title>
</head>
<body>

<h3>You made a slice</h3>

<?php

include 'example.php';

$sxe = new SimpleXMLElement($xmlstr);

if (isset($_POST['getSlice']))
{
    $nodes=$_POST['nodes'];
    $interfaces=$_POST['interfaces'];
}

```

```

$rate=$_POST['rate'];
$channel=$_POST['channel'];
$mode=$_POST['mode'];

if (isset($_GET['name'])      &&  isset($_GET['date'])      &&  isset($_GET['hour_init'])      &&
isset($_GET['duration']))
{
    $name_init = $_GET['name'];
    //$owner=$_GET['owner'];
    $date=$_GET['date'];
    $hour_init=$_GET['hour_init'];
    $duration=$_GET['duration'];
}

$name1="";
$array=split(" ", $name_init);
for ($x=0;$x<count($array);$x++){
    $name1.=$array[$x];
}

$name2="";
$array=split(" ", $name1);
for ($x=0;$x<count($array);$x++){
    $name2.=$array[$x];
}

$name3="";
$array=split(" ", $name2);
for ($x=0;$x<count($array);$x++){
    $name3.=$array[$x];
}

$name = preg_replace("/[^\a-zA-Z0-9s_]/", "", $name3);
if (isset($name)      &&  isset($date)      &&  isset($hour_init)      &&  isset($duration)      &&  ($name<>'')      &&
($owner<>''))
{
    $flag=0;
    for ($i=0;$i<count($interfaces);$i++){
        $check=$interfaces[$i];
        $array=split(" ", $check);
        $id_check=$array[0];
        if (!in_array($id_check, $nodes)) {
            $flag=1;
        }
    }
if ($flag==1)
{
    echo "You have selected interfaces but not the node.<a
href=\"http://sorch.netmode.ntua.gr/ws/check_time.php\">Please try again</a><br/><br/>";
}
else
{
    $timestamp=$date." ".$hour_init;
    $rspec=new RSpec();

    $rspec->name=$name;
    $rspec->owner=$owner;
    $rspec->timestamp=$timestamp;
    $rspec->duration=$duration;

    $y=0;
    for ($x=0;$x<count($nodes);$x++){
        $node=$sxe->addChild('node');

        $id=$nodes[$x];
        $node->addAttribute('client_id', "omf.netmode.node".$id);
        $node-
        >addAttribute('component_id', "urn:publicid:IDN+sorch.netmode.ntua.gr+node+omf.netmode.node".$id);
        $node->addAttribute('component_manager_id', "urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
        $node->addAttribute('exclusive', 'true');

        $hardware=$node->addChild('hardware_type');

        $hardware->addAttribute('name', 'wireless access point');

        $sliver=$node->addChild('sliver_type');

        $name1="omf.netmode.node".$id;
        $sliver->addAttribute('name', 'access point');

        //$z=$id*2-2;
        for ($z=0;$z<2;$z++){
            $check=$interfaces[$y];
            $array=split(" ", $check);

```

```

//echo $check;
$cid_check=$array[0];

if ($cid_check==$id)
{
$y=$y+1;
$if1=$array[1];
$i=$array[2];

$interface=$node->addChild('interface');

$if=$name1.".".$if1;
$interface->setAttribute('client_id',$if);

if (strcmp($if1,'if0')==0)
{
$m=$i*2;
$wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");

$channell=$wireless-
>addChild('RadioChannel',$channel[$m],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
$ratel=$wireless-
>addChild('Rate',$rate[$m],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
$model=$wireless-
>addChild('OperatingMode',$mode[$m],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
}
else
{
$m=$i*2+1;
$wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");

$channell=$wireless-
>addChild('RadioChannel',$channel[$m],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
$ratel=$wireless-
>addChild('Rate',$rate[$m],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
$model=$wireless-
>addChild('OperatingMode',$mode[$m],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
//$z=$z+1;
}
}
}

$string1=$sxe->asXML();

$rspec->rspec1=$string1;
$client = new SoapClient("sliceoperations.wsdl");
$response1 = $client->mkslice($rspec);

}
}
else
{
echo "You have not put right the parameters.<a href=\"http://sorch.netmode.ntua.gr/ws/check_time.php\">Please try again</a><br/><br/>";
}
}

echo $response1."<br/>";

echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
?>

</body>
</html>

```

modifyslice.php

```

<?php
//php script to select the slice that you want to modify among the existing categories
session_start();
$owner=$_SESSION['username'];

Echo "<html>";
Echo "<title>Modify a slice</title>";

$client = new SoapClient("listslices12.wsdl");
$client1 = new SoapClient("listslices12.wsdl");
$client2 = new SoapClient("listslices12.wsdl");
$response5 = $client->getListMyActivatedSlices($owner);

```

```

$response4=$response5->validatearray;
$response3 = $client1->getListMyReservedSlices($owner);
$response1 = $client2->getListMyNoReservedSlices($owner);
$response2=$response1->validatearray;
$response6=$response3->validatearray;

if((in_array(2,$response2) || (in_array(2,$response4) || (in_array(2,$response6))))
{
echo "not valid rspec";
}
else
{
$response=$response5->slicearray;
$reserved=array();
for($x = 0 ; $x < count($response) ; $x++){
$reserved[]=$response[$x]->name;
}

Echo "<form method=\"post\" action=\"http://sorch.netmode.ntua.gr/ws/modifyslice2new.php\">";
Echo "<h1>Modify a slice</h1>";
$response=$response1->slicearray;
Echo "<h3>No-activated slices</h3>";
Echo "<p><b>List of no-reserved Slices which are not activate now</b></p>";
if (count($response)==0)
{
echo "There is not any noreserved slice that is noactivated <br/>";
}
else{

for($x = 0 ; $x < count($response) ; $x++){
$name1=$response[$x]->name;
Echo "<input type=\"radio\" name=\"name\" value=$name1>The ".$response[$x]->name." with owner
".$response[$x]->owner." contains the following nodes of testbed: ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\+",$string1);
$hostname=$array1[3];

Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];
Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";

}
}
Echo "</ul>";
}

Echo "</ol>";
}
}

```

```

}

$response=$response3->slicearray;
Echo "<p><b>List of reserved Slices which are not activate now</b></p>";
if (count($response)==0)
{
echo "There is not any reserved slice that is noactivated now <br/>";
}
else{
for($x = 0 ; $x < count($response) ; $x++){
if(!in_array($response[$x]->name,$reserved)){
$time_start=$response[$x]->time_start;
$array_start=split(",",$time_start);

$time_end=$response[$x]->time_end;
$array_end=split(",",$time_end);

$name1=$response[$x]->name;
Echo "<input type=\"radio\" name=\"name\" value=$name1>The ".$response[$x]->name." with owner
".$response[$x]->owner;
Echo " has been reserved ";
for($y = 0 ; $y < count($array_start) ; $y++){

if ($y==0)
{
Echo " from ".$array_start[$y]." to ".$array_end[$y];
}
else
{
Echo " and from ".$array_start[$y]." to ".$array_end[$y];
}
}

Echo ".The slice contains the following nodes of testbed: ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\\+",$string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;

Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";

}
}
Echo "</ul>";
}
}

```

```

    Echo "</ol>";
}
}

Echo "<h3>Activated Slices</h3>";
$response=$response5->slicearray;

if (count($response)==0)
{
echo "There is not any activated slice <br/>";
}
else{
for($x = 0 ; $x < count($response) ; $x++){
$name1=$response[$x]->name;
Echo "<input type=\"radio\" name=\"name\" value=$name1>The ".$response[$x]->name." with owner
".$response[$x]->owner." ,which has been activated from ".$response[$x]->time_start." and for
".$response[$x]->duration." hours contains the following nodes of testbed: ";

$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;

Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$sif_id=$array[1];

Echo "<li>interface ".$sif_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";

}
}
Echo "</ul>";

}

Echo "</ol>";
}

}

?>
<input type="submit" value="submit" name="getSlice">
<input type="reset" value="Reset" name="reset">

</form>
<?php
}
?>

```

modifyslice2new.php

```
<?php
//php scripts for selecting the available resources in order to modify the slice that //you have
chosen
$getSlice = isset($_POST["getSlice"]) ? $_POST['getSlice'] : '';
$name = isset($_POST["name"]) ? $_POST['name'] : '';

?>

<html>
<head>
<title> Modify a slice </title>
</head>
<body>

<?php

if (isset($_POST['getSlice']))
{

if (isset($_GET['name']))
{
    $name = $_GET['name'];
}

if (isset($name))
{
    $channels=array("IEEE 802.11b/g:Channel 1:2412MHz","IEEE 802.11b/g:Channel 2:2417MHz","IEEE
802.11b/g:Channel 3:2422MHz","IEEE 802.11b/g:Channel 4:2427MHz","IEEE 802.11b/g:Channel
5:2432MHz","IEEE 802.11b/g:Channel 6:2437MHz","IEEE 802.11b/g:Channel 7:2442MHz","IEEE
802.11b/g:Channel 8:2447MHz","IEEE 802.11b/g:Channel 9:2452MHz","IEEE 802.11b/g:Channel
10:2457MHz","IEEE 802.11b/g:Channel 11:2462MHz","IEEE 802.11b/g:Channel 12:2467MHz","IEEE
802.11b/g:Channel 13:2472MHz");

    $client = new SoapClient("listslices12.wsdl");
    $client2 = new SoapClient("listslices12.wsdl");
    $response2 = $client2->getListAvailableResources($name);
    $response1 = $client->getListSliceResources($name);

    $i=0;
    $x=0;
    $node_ids=array();
    $nodes=array();
    $nodes1=array();
    $nodes2=array();
    $response3=array();
    $rates=array();
    $modes=array();

    $simplexml = new SimpleXMLElement($response1);
    foreach ($simplexml->node as $entry2){
    $nodes[]=$entry2;
    }

    $simplexml2 = new SimpleXMLElement($response2);

    for ($x=0;$x<count($nodes);$x++)
    {
        $node=$nodes[$x];
        $component_id=$node->attributes()->component_id;

        $array=split(":",$component_id);
        $string1=$array[2];

        $array1=split("\+",$string1);
        $hostname=$array1[3];

        $id=preg_replace("/^[^0-9s]/", "", $hostname);
        $node_ids[]=$id;
    }

    foreach ($simplexml2->node as $entry4){
    $node=$entry4;
    $component_id=$node->attributes()->component_id;

    $array=split(":",$component_id);
```

```

$string1=$array[2];

$array1=split("\+", $string1);
$hostname=$array1[3];

$id=preg_replace("/[^0-9s]/", "", $hostname);
if(!in_array($id,$node_ids)){
$node_ids1[]=$id;
$nodes1[]=$entry4;
}
}

foreach ($simplexml2->node as $entry5){
$nodes2[]=$entry5;
}

$node=$nodes2[$i];

foreach ($node->interface as $interfacel)
{
$interfacesarray[]=$interfacel;
}
$interface2=$interfacesarray[$i];

$dc = $interface2->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc->Wireless_features as $entry1){
$dc1 = $entry1->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1->RadioChannel as $entry3){
$response3[]=$entry3;
}
foreach ($dc1->Rate as $entry3){
$rate=$entry3;
}
foreach ($dc1->OperatingMode as $entry3){
$modes[]=$entry3;
}
}

$ratel=split(" ", $rate);
for ($i=0;$i<count($ratel);$i++)
{
$rates[]=$ratel[$i];
}

$nodes_all=array();
if (count($node_ids)==0){
$nodes_all=$node_ids1;
}

if (count($node_ids1)==0){
$nodes_all=$node_ids;
}

if ((count($node_ids)<>0) && (count($node_ids1)<>0))
{
$nodes_all=array_merge($node_ids,$node_ids1);
sort($nodes_all);
}

Echo "<h1>Modify a slice</h1>"; ?>
Choose the nodes and the wireless features that you want to add to the slice <?php echo $name; ?>
:<br><br>

<?php
Echo "<form method=\"post\" action=\"http://sorch.netmode.ntua.gr/ws/ modifyslice3. php\">";
$m=0;
$k=0;
$l=0;
echo $number_nodes;
for($x = 0 ; $x < count($nodes_all) ; $x++){
$node_check=$nodes_all[$x];
if (in_array($node_check,$node_ids)){
$component_id=$nodes[$m]->attributes()->component_id;

$array=split(":", $component_id);
$string1=$array[2];

$array1=split("\+", $string1);
$hostname=$array1[3];

```



```

$cid=preg_replace("/[^\d-]/", "", $hostname);

Echo"<b><input type=\"checkbox\" name=\"nodes[]\" value=$cid checked> $hostname <br></b>";
$interfacesarray=array();
foreach ($nodes[$m]->interface as $interface1)
{
$interfacesarray[]=$interface1;
}
$m=$m+1;
$if_id=array();

for ($q=0;$q<count($interfacesarray);$q++){
$interface2=$interfacesarray[$q];
$component_id2=$interface2['client_id'];
$array=split(':', $component_id2);
$if_id[]=$array[1];
}

$q=0;
for ($y=0;$y<2;$y++)
{

$temp="if".$y;
$if=$cid."".$temp."".$s1;

if (in_array($temp,$if_id) {
$interface2=$interfacesarray[$q];
Echo" <input type=\"checkbox\" name=\"interfaces[]\" value=$if checked> $hostname $temp <br>";
$q=$q+1;
$dc = $interface2->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc->Wireless_features as $entry1){
$dcl = $entry1->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");

$rate=$entry1->Rate;
$channel1=$entry1->RadioChannel;
$mode=$entry1->OperatingMode;

for ($it1=0;$it1<count($rates);$it1++){

if (strcmp($rate,$rates[$it1])==0) {

if ($it1==0) {
?>

<select name="rate[]">
<option value="<?php echo $rates[$it1] ?>" selected="selected"><?php echo $rates[$it1] ?>
Mbps</option>

<?php
}
else
{
?>
<option value="<?php echo $rates[$it1] ?>" selected="selected"><?php echo $rates[$it1] ?>
Mbps</option>
<?php
}
}
else
{
if ($it1==0) {
?>

<select name="rate[]">
<option value="<?php echo $rates[$it1] ?>"><?php echo $rates[$it1] ?> Mbps</option>

<?php
}
else
{
?>
<option value="<?php echo $rates[$it1] ?>"><?php echo $rates[$it1] ?> Mbps</option>
<?php
}
}
}
?>
</select>

<?php
//$it2=0;
for ($z=0;$z<count($response3);$z++){
$temp1=$response3[$z];
if (strcmp($channel1,$temp1)==0) {

if ($z==0) {

```

```

?>

<select name="channel[]">
  <option value="<?php echo $temp1 ?>" selected="selected"><?php echo $channels[$temp1-1] ?>
</option>

<?php
}
else
{
?>
<option value="<?php echo $temp1 ?>" selected="selected"><?php echo $channels[$temp1-1] ?>
</option>
<?php
}
}
else
{
if ($z==0){
?>

<select name="channel[]">
  <option value="<?php echo $temp1 ?>"><?php echo $channels[$temp1-1] ?> </option>

<?php
}
else
{
?>
<option value="<?php echo $temp1 ?>"><?php echo $channels[$temp1-1] ?> </option>
<?php
}
}
//$it2=$it2+1;
}
?>
</select>

<?php
for ($it3=0;$it3<count($modes);$it3++){

if (strcmp($mode,$modes[$it3])==0){

if ($it3==0){
?>

<select name="mode[]">
  <option value="<?php echo $modes[$it3] ?>" selected="selected"><?php echo $modes[$it3]
?></option>

<?php
}
else
{
?>
<option value="<?php echo $modes[$it3] ?>" selected="selected"><?php echo $modes[$it3] ?>
</option>
<?php
}
}
else
{
if ($it3==0){
?>

<select name="mode[]">
  <option value="<?php echo $modes[$it3] ?>"><?php echo $modes[$it3] ?> </option>

<?php
}
else
{
?>
<option value="<?php echo $modes[$it3] ?>"><?php echo $modes[$it3] ?> </option>
<?php
}
}
?>
</select>
</br>

<?php
}
}
else
{
Echo " <input type="checkbox\" name=\"interfaces[]\" value=$if> $hostname $temp <br>";

```

```

?>
<select name="rate[]">
  <?php
  for ($z=0;$z<count($rates);$z++){
  $temp=$rates[$z];
  ?>
  <option value="<?php echo $temp ?>"><?php echo $temp ?> Mbps</option>
  <?php
  }
  ?>
</select>

<select name="channel[]">
  <?php
  for ($z=0;$z<count($response3);$z++){
  $temp=$response3[$z];
  ?>
  <option value="<?php echo $temp ?>"><?php echo $channels[$temp-1] ?> </option>
  <?php
  }
  ?>
</select>

<select name="mode[]">
  <?php
  for ($z=0;$z<count($modes);$z++){
  $temp=$modes[$z];
  ?>
  <option value="<?php echo $temp ?>"><?php echo $temp ?></option>
  <?php
  }
  ?>
</select>
<br/>
<?php
}
}
$1=$1+1;
Echo"<br/>";
}
else
{
if (in_array($node_check,$node_ids1)){
$node1=$nodes1[$k];
$component_id=$node1->attributes()->component_id;

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\\+", $string1);
$hostname=$array1[3];

$id=preg_replace("/[^0-9s]/", "", $hostname);

Echo"<b><input type=\"checkbox\" name=\"nodes[]\" value=$id> $hostname <br/></b>";

foreach ($node1->interface as $interface1)
{
$component_id2=$interface1['component_name'];
$array=split(':', $component_id2);
$if=$id.".". $array[1].".".$1;
Echo" <input type=\"checkbox\" name=\"interfaces[]\" value=$if> $hostname $array[1] <br>";
?>
<select name="rate[]">
  <?php
  for ($z=0;$z<count($rates);$z++){
  $temp=$rates[$z];
  ?>
  <option value="<?php echo $temp ?>"><?php echo $temp ?> Mbps</option>
  <?php
  }
  ?>
</select>

<select name="channel[]">
  <?php
  for ($z=0;$z<count($response3);$z++){
  $temp=$response3[$z];
  ?>
  <option value="<?php echo $temp ?>"><?php echo $channels[$temp-1] ?> </option>
  <?php
  }
  ?>
</select>

<select name="mode[]">

```

```

    <?php
    for ($z=0;$z<count($modes);$z++){
    $temp=$modes[$z];
    ?>
    <option value="<?php echo $temp ?>"><?php echo $temp ?></option>
    <?php
    }
    ?>
</select>
<br/>
<?php
}
Echo"<br/>";
$k=$k+1;
$l=$l+1;
}
}
}
Echo "<input type=\"hidden\" name=\"name\" value=$name />";
?>

<br/>
<input type="submit" value="submit" name="getSlice">
<input type="reset" value="Reset" name="reset">
<?php
}
}
?>
</form>

</html>

```

modifyslice3.php

```

<?php
//php script for calling the operation modifyslice in order to modify the specific //slice with
the new parameters
$getSlice = isset($_POST["getSlice"]) ? $_POST['getSlice'] : '';
$nodes = isset($_POST["nodes"]) ? $_POST['nodes'] : '';
$interfaces= isset($_POST["interfaces"]) ? $_POST['interfaces'] : '';
$rate=isset($_POST["rate"]) ? $_POST['rate'] : '';
$mode=isset($_POST["mode"]) ? $_POST['mode'] : '';
$channel=isset($_POST["channel"]) ? $_POST['channel'] : '';
$name=isset($_POST["name"]) ? $_POST['name'] : '';

class RSpec {
public $name;
public $owner;
public $timestamp;
public $duration;
public $rspec1;
}

?>

<html>
<head>
<title>Modify a slice</title>
</head>
<body>

<h3>You modified the slice <?php echo $name; ?></h3>

<?php

include 'example.php';

$sxe = new SimpleXMLElement($xmlstr);

if (isset($_POST['getSlice']))
{
    $nodes=$_POST['nodes'];
    $interfaces=$_POST['interfaces'];
    $rate=$_POST['rate'];
    $channel=$_POST['channel'];
    $mode=$_POST['mode'];

    if (isset($_GET['name']))
    {
        $name = $_GET['name'];
    }

    if (isset($name) && ($name<>''))
    {

```

```

$flag=0;
for($i=0;$i<count($interfaces);$i++){
    $check=$interfaces[$i];
    $array=split(",",$check);
    $id_check=$array[0];
    if (!in_array($id_check,$nodes)){
        $flag=1;
    }
}
if ($flag==1)
{
    echo "You have selected interfaces but not the node.<a
href=\"http://sorch.netmode.ntua.gr/ws/modifyslice.php\">Please try again</a><br/> <br/>";
}
else
{
    $rspec=new RSpec();

    $rspec->name=$name;
    $y=0;
    for ($x=0;$x<count($nodes);$x++){
        $node=$sxe->addChild('node');

        $id=$nodes[$x];
        $node->addAttribute('client_id',"omf.netmode.node".$id);
        $node-
        >addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+node+omf.netmode.node".$id);
        $node->addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
        $node->addAttribute('exclusive','true');

        $hardware=$node->addChild('hardware_type');

        $hardware->addAttribute('name','wireless access point');

        $sliver=$node->addChild('sliver_type');

        $name1="omf.netmode.node".$id;
        $sliver->addAttribute('name','access point');

        for ($z=0;$z<2;$z++){
            $check=$interfaces[$y];
            $array=split(",",$check);

            $id_check=$array[0];

            if ($id_check==$id)
            {
                $y=$y+1;
                $if1=$array[1];
                $i=$array[2];

                $interface=$node->addChild('interface');

                $if=$name1.".".$if1;
                $interface->addAttribute('client_id',$if);

                if (strcmp($if1,'if0')==0)
                {

                    $m=$i*2;
                    $wireless=$interface-
                    >addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");

                    $channell=$wireless-
                    >addChild('RadioChannel',$channel[$m],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
                    $ratel=$wireless-
                    >addChild('Rate',$rate[$m],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
                    $model=$wireless-
                    >addChild('OperatingMode',$mode[$m],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
                }
            }
            else
            {

                $m=$i*2+1;
                $wireless=$interface-
                >addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");

                $channell=$wireless-
                >addChild('RadioChannel',$channel[$m],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
                $ratel=$wireless-
                >addChild('Rate',$rate[$m],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
                $model=$wireless-
                >addChild('OperatingMode',$mode[$m],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
            }
        }
    }
}
}

```

```

}

$string1=$sxe->asXML();

$rspec->rspec1=$string1;

$client = new SoapClient("sliceoperations.wsdl");
$response1 = $client->modifyslice($rspec);

}
}
else
{
echo "You have not put right the parameters.<a href=\"http://sorch.netmode.ntua.gr/ws/mkslicenew.php\">Please try again</a> <br/> <br/>";
}
}

echo $response1."<br/>";

echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
?>

</body>
</html>

```

reservesliceservice.php

```
<?php
```

```

//function for reserving a slice of user for a specific time in the future if the //shared
resources (nodes and frequency channels) are not being used by another //activated slice
function reserveslice($message){

$simplexml = new SimpleXMLElement($message->str);

$name =$simplexml->name;
$user1=$simplexml->user;
$duration=$simplexml->duration;
$date=$simplexml->date;
$hour_init=$simplexml->hour_init;

$timestamp=$date." ".$hour_init;

$array = split("-", $date);

$host = "localhost";
$user = "omf";
$pass = "omf";
$database = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($database, $linkID) or die("Could not find database.");

$query="SELECT * FROM slices WHERE name='$name'";

$result = mysql_query($query, $linkID) or die("Data not found.");
$row = mysql_fetch_assoc($result);
if ($row<1)
{
$responsePayloadString = <<<XML
<reservesliceresponse>
<result>This slice with name $name does no exist so try activate another slice</result>
</reservesliceresponse>
XML;

$outMessage = new WSMMessage($responsePayloadString);

return $outMessage;
}
else
{

$querytimestamp="SELECT UNIX_TIMESTAMP('$timestamp')";
$resulttimestamp=mysql_query($querytimestamp, $linkID);
$rowtimestamp = mysql_fetch_assoc($resulttimestamp);
$timestamp_start=$rowtimestamp["UNIX_TIMESTAMP('$timestamp')"];

$time=time();

if ((checkdate($array[1],$array[2],$array[0])) && ($timestamp_start>=$time))
{

$duration_timestamp=3600*$duration;

$timestamp_end=$timestamp_start+$duration_timestamp;

```

```

$query_check1="SELECT * FROM reservations WHERE name='$name'";

$result_check1 = mysql_query($query_check1, $linkID) or die("Data not found.");
$num_rows=mysql_num_rows($result_check1);

$check1=array();

$check_string="";
if ($num_rows<1)
{
$check1[]=1;
}
else
{
$warning_message="";
for($x = 0 ; $x < mysql_num_rows($result_check1) ; $x++){
    $row_check1 = mysql_fetch_assoc($result_check1);

    $timestamp_start_check=$row_check1['time_activation_start'];
    $timestamp_end_check=$row_check1['time_activation_end'];

$querytimestamp1="SELECT UNIX_TIMESTAMP('$timestamp_start_check')";
$resulttimestamp1=mysql_query($querytimestamp1, $linkID);
$rowtimestamp1 = mysql_fetch_assoc($resulttimestamp1);
$timestamp_start1=$rowtimestamp1["UNIX_TIMESTAMP('$timestamp_start_check')"];

$querytimestamp2="SELECT UNIX_TIMESTAMP('$timestamp_end_check')";
$resulttimestamp2=mysql_query($querytimestamp2, $linkID);
$rowtimestamp2= mysql_fetch_assoc($resulttimestamp2);
$timestamp_end1=$rowtimestamp2["UNIX_TIMESTAMP('$timestamp_end_check')"];

$check4=($timestamp_start<=$timestamp_start1) && ($timestamp_end<=$timestamp_start1);
$check5=($timestamp_start>=$timestamp_end1) && ($timestamp_end>=$timestamp_end1);

if ($check4 || $check5)
{
$check1[$x]=1;
}
else
{
$check1[$x]=2;
$warning_message=$warning_message."The slice ".$name." has been activated from
".$timestamp_start_check." to ".$timestamp_end_check." .Please choose another hour \n";
}
}

if (!in_array(2,$check1))
{
$warning_message2="";
$check2=array();
$query_check2="SELECT slice_id FROM slices WHERE name='$name'";
$result_check2 = mysql_query($query_check2, $linkID) or die("Data not found.");
$row_check2 = mysql_fetch_assoc($result_check2);
$slice_id_check=$row_check2['slice_id'];

$query_check3="SELECT node_id FROM slice_node association WHERE slice_id={ $slice_id_check}";
$result_check3 = mysql_query($query_check3, $linkID) or die("Data not found.");
$num_rows2=mysql_num_rows($result_check3);

if ($num_rows2<1)
{
$check2[]=1;
}
else
{
for($x = 0 ; $x < mysql_num_rows($result_check3) ; $x++){
    $row_check3= mysql_fetch_assoc($result_check3);
    $node_id_check=$row_check3['node_id'];

    $query_check4="SELECT slice_id FROM slice_node_association WHERE slice_id<> { $slice_id_check}
AND node_id={ $node_id_check}";
    $result_check4 = mysql_query($query_check4, $linkID) or die("Data not found.");
    $num_rows3=mysql_num_rows($result_check4);
    if ($num_rows3<1)
    {
        $check2[]=1;
    }
    else
    {
        for($y = 0 ; $y < mysql_num_rows($result_check4) ; $y++){
            $row_check4= mysql_fetch_assoc($result_check4);
            $slice_id_check1=$row_check4['slice_id'];

            $query_check5="SELECT * FROM slices WHERE slice_id={ $slice_id_check1}";
            $result_check5 = mysql_query($query_check5, $linkID) or die("Data not found.");

```

```

$row_check5= mysql_fetch_assoc($result_check5);
$name_check=$row_check5['name'];
$owner=$row_check5['owner'];

$query_check6="SELECT * FROM reservations WHERE name='$name_check'";

$result_check6 = mysql_query($query_check6, $linkID) or die("Data not found.");
$num_rows4=mysql_num_rows($result_check6);

$check3=array();

if ($num_rows4<1)
{
    $check2[]=1;
}
else
{
for($z = 0 ; $z < mysql_num_rows($result_check6) ; $z++){
    $row_check6 = mysql_fetch_assoc($result_check6);

    $timestamp_start_check2=$row_check6['time_activation_start'];
    $timestamp_end_check2=$row_check6['time_activation_end'];

$querytimestamp3="SELECT UNIX_TIMESTAMP('$timestamp_start_check2')";
$resulttimestamp3=mysql_query($querytimestamp3, $linkID);
$rowtimestamp3 = mysql_fetch_assoc($resulttimestamp3);
$timestamp_start2=$rowtimestamp3["UNIX_TIMESTAMP('$timestamp_start_check2')"];

$querytimestamp4="SELECT UNIX_TIMESTAMP('$timestamp_end_check2')";
$resulttimestamp4=mysql_query($querytimestamp4, $linkID);
$rowtimestamp4= mysql_fetch_assoc($resulttimestamp4);
$timestamp_end2=$rowtimestamp4["UNIX_TIMESTAMP('$timestamp_end_check2')"];

$check6=($timestamp_start<=$timestamp_start2) && ($timestamp_end<=$timestamp_start2);
$check7=($timestamp_start>=$timestamp_end2) && ($timestamp_end>=$timestamp_end2);

if ($check6 || $check7)
{
    $check3[]=1;
}
else
{
    $check3[]=2;
    $warning_message2="The node ".$node_id_check." is used from the slice ".$name_check." with owner
    ".$owner." which has been activated from ".$timestamp_start_check2." to
    ".$timestamp_end_check2.",";
}
}

if (in_array(2,$check3))
{
    $check2[]=2;
}
else
{
    $check2[]=1;
}
}
}
}

if (!in_array(2,$check2))
{
$querytimestamp_final1="SELECT FROM_UNIXTIME('$timestamp_start')";
$resulttimestamp_final1=mysql_query($querytimestamp_final1, $linkID);
$rowtimestamp_final1 = mysql_fetch_assoc($resulttimestamp_final1);
$timestamp_start_final=$rowtimestamp_final1["FROM_UNIXTIME('$timestamp_start')"];

$querytimestamp_final2="SELECT FROM_UNIXTIME('$timestamp_end')";
$resulttimestamp_final2=mysql_query($querytimestamp_final2, $linkID);
$rowtimestamp_final2 = mysql_fetch_assoc($resulttimestamp_final2);
$timestamp_end_final=$rowtimestamp_final2["FROM_UNIXTIME('$timestamp_end')"];

$event1=$name."_2".$timestamp_start;
$query1="CREATE EVENT $event1
ON SCHEDULE AT '$timestamp_start_final'
DO
UPDATE inventory2.slices SET
activate=1,duration_activation='$duration',timeactivation='$timestamp_start_final' WHERE
name='$name'";

mysql_query($query1, $linkID);
$query3="INSERT INTO inventory2.reservations (`name`,`time_activation_start`,
`time_activation_end`) VALUES ('$name','$timestamp_start_final','$timestamp_end_final')";

```



```

mysql_query($query3, $linkID);

$event=$name."_".$timestamp_start;
$query2="CREATE EVENT $event
        ON SCHEDULE AT '$timestamp_start_final' + INTERVAL '$duration' HOUR
        DO
        UPDATE inventory2.slices SET activate=0,duration_activation='0', timeactivation='0000-00-
00 00:00:00' WHERE name='$name'";

mysql_query($query2, $linkID);

$responsePayloadString = <<<XML
<reservesliceresponse>
<result>$user1 has reserved the slice $name at $timestamp_start_final for $duration hours</result>
</reservesliceresponse>
XML;

$outMessage = new WSMMessage($responsePayloadString);

return $outMessage;
}
else
{
$responsePayloadString = <<<XML
<reservesliceresponse>
<result>$warning_message2</result>
</reservesliceresponse>
XML;

$outMessage = new WSMMessage($responsePayloadString);

return $outMessage;
}
}
else
{
$responsePayloadString = <<<XML
<reservesliceresponse>
<result>$warning_message</result>
</reservesliceresponse>
XML;

$outMessage = new WSMMessage($responsePayloadString);

return $outMessage;
}
}
else
{
$responsePayloadString = <<<XML
<reservesliceresponse>
<result>You have not put right the date you want to use the slice $name</result>
</reservesliceresponse>
XML;

$outMessage = new WSMMessage($responsePayloadString);

return $outMessage;
}
}
}

$operations = array("reserveslice"=>"reserveslice");

$service = new WSService(array ("operations" => $operations));
// process client requests and reply

$service->reply();
?>

```

reserveslice.php

```

<?php
//php script for selecting the slice that you want to reserve and the time of the //reservation

session_start();
$owner=$_SESSION['username'];

Echo "<html>";
Echo "<title>Reserve a slice</title>";

$client = new SoapClient("listslices12.wsdl");
$client1 = new SoapClient("listslices12.wsdl");

```

```

$client2 = new SoapClient("listslices12.wsdl");
$response5 = $client->getListMyActivatedSlices($owner);
$response4=$response5->validatearray;
$response3 = $client1->getListMyReservedSlices($owner);
$response1 = $client2->getListMyNoReservedSlices($owner);
$response2=$response1->validatearray;
$response6=$response3->validatearray;

if((in_array(2,$response2)) || (in_array(2,$response4)) || (in_array(2,$response6)))
{
echo "not valid rspec";
}
else
{
$response=$response5->slicearray;
$reserved=array();
for($x = 0 ; $x < count($response) ; $x++){
$reserved[]=$response[$x]->name;
}

Echo "<form method=\"post\" action=\"http://sorch.netmode.ntua.gr/ws/reserveslice2.php\">";
Echo "<h1>Reserve a slice</h1>";
$response=$response1->slicearray;
Echo "<h3>No-activated slices</h3>";
Echo "<p><b>List of no-reserved Slices which are not activate now</b></p>";
if (count($response)==0)
{
echo "There is not any noreserved slice that is noactivated <br/>";
}
else{
for($x = 0 ; $x < count($response) ; $x++){
$name1=$response[$x]->name;
Echo "<input type=\"radio\" name=\"name\" value=$name1>The ".$response[$x]->name." with owner
".$response[$x]->owner." contains the following nodes of testbed: ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;

Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)

```

```

{
Echo " Mbps ";
}
}
}
Echo "</li>";

}
}
Echo "</ul>";
}

Echo "</ol>";
}
}

$response=$response3->slicearray;
Echo "<p><b>List of reserved Slices which are not activate now</b></p>";
if (count($response)==0)
{
echo "There is not any reserved slice that is noactivated now <br/><br/>";
}
else{
for($x = 0 ; $x < count($response) ; $x++){
if(!in_array($response[$x]->name,$reserved)){
$time_start=$response[$x]->time_start;
$array_start=split(",",$time_start);

$time_end=$response[$x]->time_end;
$array_end=split(",",$time_end);

$name1=$response[$x]->name;
Echo "<input type=\"radio\" name=\"name\" value=$name1>The ".$response[$x]->name." with owner
".$response[$x]->owner;
Echo " has been reserved ";
for($y = 0 ; $y < count($array_start) ; $y++){

if ($y==0)
{
Echo " from ".$array_start[$y]." to ".$array_end[$y];
}
else
{
Echo " and from ".$array_start[$y]." to ".$array_end[$y];
}
}

Echo ".The slice contains the following nodes of testbed: ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;

Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{

```

```

$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";

}
}
Echo "</ul>";
}

Echo "</ol>";
}
}

Echo "<h3>Activated Slices</h3>";
$response=$response5->slicearray;

if (count($response)==0)
{
echo "There is not any activated slice <br/><br/>";
}
else{
for($x = 0 ; $x < count($response) ; $x++){
$name1=$response[$x]->name;
Echo "<input type=\"radio\" name=\"name\" value=$name1>The ".$response[$x]->name." with owner
".$response[$x]->owner." ,which has been activated from ".$response[$x]->time_start." and for
".$response[$x]->duration." hours contains the following nodes of testbed: ";

$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;

Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check1=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);

```

```

$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interface1->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
//$iterator='yes';
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";

}
}
Echo "</ul>";
}

Echo "</ol>";
}

}

?>

<?php
}

?>

Put the date that you want to activate the slice:

<select name="month">
  <option> - Month - </option>
  <option value="01">January</option>
  <option value="02">Febuary</option>
  <option value="03">March</option>
  <option value="04">April</option>
  <option value="05">May</option>
  <option value="06">June</option>
  <option value="07">July</option>
  <option value="08">August</option>
  <option value="09">September</option>
  <option value="10">October</option>
  <option value="11">November</option>
  <option value="12">December</option>
</select>

/<select name="day">
  <option> - Day - </option>
  <option value="01">1</option>
  <option value="02">2</option>
  <option value="03">3</option>
  <option value="04">4</option>
  <option value="05">5</option>
  <option value="06">6</option>
  <option value="07">7</option>
  <option value="08">8</option>
  <option value="09">9</option>
  <option value="10">10</option>
  <option value="11">11</option>
  <option value="12">12</option>
  <option value="13">13</option>
  <option value="14">14</option>
  <option value="15">15</option>
  <option value="16">16</option>
  <option value="17">17</option>

```

```

        <option value="18">18</option>
        <option value="19">19</option>
        <option value="20">20</option>
        <option value="21">21</option>
        <option value="22">22</option>
        <option value="23">23</option>
        <option value="24">24</option>
        <option value="25">25</option>
        <option value="26">26</option>
        <option value="27">27</option>
        <option value="28">28</option>
        <option value="29">29</option>
        <option value="30">30</option>
        <option value="31">31</option>
    </select>

    /<select name="year">
        <option> - Year - </option>
        <option value="2011">2011</option>
        <option value="2012">2012</option>
        <option value="2013">2013</option>
        <option value="2014">2014</option>
        <option value="2015">2015</option>
        <option value="2016">2016</option>
        <option value="2017">2017</option>
        <option value="2018">2018</option>
        <option value="2019">2019</option>
        <option value="2020">2020</option>
    </select>
<br/>

```

Put the hour that you want to activate the slice:

```

<select name="hour_start">
    <option> - Start Hour - </option>
    <option value="00">0 am</option>
    <option value="01">1 am</option>
    <option value="02">2 am</option>
    <option value="03">3 am</option>
    <option value="04">4 am</option>
    <option value="05">5 am</option>
    <option value="06">6 am</option>
    <option value="07">7 am</option>
    <option value="08">8 am</option>
    <option value="09">9 am</option>
    <option value="10">10 am</option>
    <option value="11">11 am</option>
    <option value="12">12 pm</option>
    <option value="13">1 pm</option>
    <option value="14">2 pm</option>
    <option value="15">3 pm</option>
    <option value="16">4 pm</option>
    <option value="17">5 pm</option>
    <option value="18">6 pm</option>
    <option value="19">7 pm</option>
    <option value="20">8 pm</option>
    <option value="21">9 pm</option>
    <option value="22">10 pm</option>
    <option value="23">11 pm</option>
</select>
<br />

```

Put the time that you want to use the slice in hours:<input type="text" size="2" maxlength="2" name="hour">

```

<br />

<input type="submit" value="submit" name="getSlice">
<input type="reset" value="Reset" name="reset">

</form>
</html>

```

reserveslice2.php

```

<?php
//confirmation of slice's reservation

```

```

$getSlice = isset($_POST["getSlice"]) ? $_POST['getSlice'] : '';
$name = isset($_POST["name"]) ? $_POST['name'] : '';
$user = isset($_POST["user"]) ? $_POST['user'] : '';
$hour=isset($_POST["hour"]) ? $_POST['hour'] : '';
$month=isset($_POST["month"]) ? $_POST['month'] : '';
$year=isset($_POST["year"]) ? $_POST['year'] : '';
$day=isset($_POST["day"]) ? $_POST['day'] : '';
$hour_start=isset($_POST["hour_start"]) ? $_POST['hour_start'] : '';
session_start();
$owner=$_SESSION['username'];

?>

<html>
<head>
<title>Reserve a slice</title>
</head>
<body>

<h3>Reserve a slice</h3>

<?php
if (isset($_POST['getSlice']))
{
    if ( isset($_GET['name']) && isset($_GET['hour']) && isset($_GET['hour_start']) &&
isset($_GET['month']) && isset($_GET['day']) && isset($_GET['year']))
    {
        $name = $_GET['name'];
        //$user=$_GET['user'];
        $hour=$_GET['hour'];
        $hour_start=$_GET['hour_start'];
        $day=$_GET['day'];
        $month=$_GET['month'];
        $year=$_GET['year'];
    }

    $duration=$hour;

    $date=$year."-".$month."-".$day;

    $hour_init=$hour_start.":00:00";

    //$timestamp=$year."-".$month."-".$day." ".$hour_start.":00:00";

    if ( ($hour=='') || ($year=='') || ($month=='') )
    {
        echo "You have not put all the parameters.<a
href=\"http://sorch.netmode.ntua.gr/ws/activateslice.php\">Please try again</a><br/><br/>";
    }
    else
    {
        try {
            //$name=$_GET['name'];
            //$user=$_GET['user'];
            //$duration=$_GET['duration'];
            //for ($i=0; $i<count($array); $i++){
            //echo $array[$i];
            //}

            $client = new WSClient(array("to" =>
"http://sorch.netmode.ntua.gr/ws/reservesliceservice2.php","useSOAP" => 1.1));

            $reqPayloadString1 = <<<XML
<ns1:reserveslice xmlns:ns1="http://sorch.netmode.ntua.gr/ws/reservesliceservice2.php">
<name>$name</name><user>$owner</user><duration>$duration</duration><date>$date</date><hour_init>$h
our_init</hour_init>
</ns1:reserveslice>
XML;

            $reqMessage1 = new WSMMessage($reqPayloadString1);

            $response1 = $client->request($reqMessage1);
            $simplexml1 = new SimpleXMLElement($response1->str);

            $string=$simplexml1->result;

```

```

$array=split(";", $string);

$string_node=$array[0];
$array2=split(" ", $string_node);

for ($y=0; $y<count($array2); $y++)
{
echo $array2[$y]."<br/>";
}

$string_channel=$array[1];
$array3=split(" ", $string_channel);

$array_temp=array_count_values($array3);

foreach ($array_temp as $key=>$channel)
{
echo $key."<br/>";
}

} catch (Exception $e) {

    if ($e instanceof WSFault) {
        printf("Soap Fault: %s\n", $e->Reason);
    } else {
        printf("Message = %s\n", $e->getMessage());
    }
}
}
echo "<a href=\"http://sorch.netmode.ntua.gr/ws/reserveslice.php\">Reserve another
slice</a><br/></br>";
echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
}
?>

</body>
</html>

```

deactivateservice.php

```

<?php

//function for deactivating an activated slice of user
function deactivateslice($message){

$simplexml = new SimpleXMLElement($message->str);

$name = $simplexml->name;

$host = "localhost";
$user = "omf";
$pass = "omf";
$database = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($database, $linkID) or die("Could not find database.");

$query_check1="SELECT * FROM reservations WHERE name='$name'";

$result_check1 = mysql_query($query_check1, $linkID) or die("Data not found.");
$num_rows=mysql_num_rows($result_check1);

for($x = 0 ; $x < mysql_num_rows($result_check1) ; $x++){
    $row_check1 = mysql_fetch_assoc($result_check1);

    $timestamp_start_check=$row_check1['time_activation_start'];
    $timestamp_end_check=$row_check1['time_activation_end'];

$querytimestamp1="SELECT UNIX_TIMESTAMP('$timestamp_start_check')";
$resulttimestamp1=mysql_query($querytimestamp1, $linkID);
$rowtimestamp1 = mysql_fetch_assoc($resulttimestamp1);
$timestamp_start1=$rowtimestamp1["UNIX_TIMESTAMP('$timestamp_start_check')"];

$querytimestamp2="SELECT UNIX_TIMESTAMP('$timestamp_end_check')";

```



```

$resulttimestamp2=mysql_query($querytimestamp2, $linkID);
$rowtimestamp2= mysql_fetch_assoc($resulttimestamp2);
$timestamp_end1=$rowtimestamp2["UNIX_TIMESTAMP('$timestamp_end_check')"];

$time=time();

if (($timestamp_start1<=$time) && ($timestamp_end1>=$time))
{
$event1=$name."_2".$timestamp_start1;

$query1="DROP EVENT $event1";
mysql_query($query1, $linkID);

$event=$name."_".$timestamp_start1;
$query2 ="DROP EVENT $event";
mysql_query($query2, $linkID);

$querytimestamp_final2="SELECT FROM_UNIXTIME('$time')";
$resulttimestamp_final2=mysql_query($querytimestamp_final2, $linkID);
$rowtimestamp_final2 = mysql_fetch_assoc($resulttimestamp_final2);
$timestamp_end_final=$rowtimestamp_final2["FROM_UNIXTIME('$time')"];

$query5="UPDATE reservations SET time_activation_end='$timestamp_end_final' WHERE name='$name'
AND time_activation_start='$timestamp_start_check'";
mysql_query($query5, $linkID);

$query4="UPDATE slices SET activate=0,duration_activation='0',timeactivation='0000-00-00 00:00:00'
WHERE name='$name'";
mysql_query($query4, $linkID);
}
}

$responsePayloadString = <<<XML
<activatesliceresponse>
<result>You have deactivated the slice $name</result>
</activatesliceresponse>
XML;

$outMessage = new WSMMessage($responsePayloadString);

return $outMessage;
}

$operations = array("deactivateslice"=>"deactivateslice");

$service = new WSService(array ("operations" => $operations));
// process client requests and reply

$service->reply();
?>

```

deactivate.php

```

<?php
//choose the activated slice that you want to deactivate
session_start();
$owner=$_SESSION['username'];

Echo "<html>";
Echo "<title>Deactivate a slice</title>";

$client = new SoapClient("listslices12.wsdl");
$response1 = $client->getListMyActivatedSlices($owner);
$response2=$response1->validatearray;

if(in_array(2,$response2))
{
echo "not valid rspec";
}
else
{

Echo "<h1>Deactivate a slice</h1>";
Echo "<form method=\"post\" action=\"http://sorch.netmode.ntua.gr/ws/deactivate2.php\">";

```

```

$response=$response1->slicearray;

if (count($response)==0)
{
echo "There is not any activated slice <br/>";
echo "<a href='\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
}
else{

for($x = 0 ; $x < count($response) ; $x++){
$name1=$response[$x]->name;
Echo "<input type='\"checkbox\"' name='\"name[]\"' value=$name1>The ".$response[$x]->name." with owner
".$response[$x]->owner." ,which has been activated from ".$response[$x]->time_start." and for
".$response[$x]->duration." hours contains the following nodes of testbed: </li> ";

$rspec=$response[$x]->rspec;

$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;

Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){

$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}

}
Echo "</li>";

}
}
Echo "</ul>";
}
}

```

```

    Echo "</ol>";
}

?>
<input type="submit" value="submit" name="getSlice">
<input type="reset" value="Reset" name="reset">

</form>

</html>
<?php
}
}
?>

```

deactivate2.php

```

<?php
//call the operation deactivate of deactivateservice
$getSlice = isset($_POST["getSlice"]) ? $_POST['getSlice'] : '';
$name = isset($_POST["name"]) ? $_POST['name'] : '';

?>

<html>
<head>
<title> DEACTIVATE A SLICE OF TESTBED </title>
</head>
<body>

<?php

if (isset($_POST['getSlice']))
{
    $names=$_POST['name'];
    $string_names='';
    $number_names=count($names);

    for($x = 0 ; $x < $number_names ; $x++){

        $string_names=$string_names.$names[$x];
        if ($x<($number_names-1))
        {
            $string_names=$string_names.',';
        }
    }
}

if ((count($names)!=0))
{
    try {

        $client = new WSClient(array("to" =>
"http://sorch.netmode.ntua.gr/ws/deactivateservice.php", "useSOAP" => 1.1));

        $array = split(",",$string_names);
        for ($i=0; $i<count($array); $i++){
            $name=$array[$i];

            $reqPayloadString1 = <<<XML
<ns1:deactivateslice xmlns:ns1="http://sorch.netmode.ntua.gr/ws/deactivateservice.php">
<name>$name</name>
</ns1:deactivateslice>
XML;

            $reqMessage1 = new WSMMessage($reqPayloadString1);

            $response1 = $client->request($reqMessage1);
            $simplexml1 = new SimpleXMLElement($response1->str);

            echo $simplexml1->result."<br/>";
        }
    }
}

```

```

} catch (Exception $e) {

    if ($e instanceof WSFault) {
        printf("Soap Fault: %s\n", $e->Reason);
    } else {
        printf("Message = %s\n", $e->getMessage());
    }
}
}
}
else
{
echo "You have not deactivated any slice </br>";
}
echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
}
?>

</body>
</html>

```

cancelreservations.php

```

<?php
//function cancelreservations in order to cancel a reservation of a slice
function cancelreservations($message){

$simplexml = new SimpleXMLElement($message->str);

$name1 = $simplexml->name;

$host = "localhost";
$user = "omf";
$pass = "omf";
$database = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($database, $linkID) or die("Could not find database.");

$array=split(";", $name1);
$name=$array[0];
$timestamp1=$array[1];
$timestamp2=$array[2];

$timestamp=$timestamp1." ".$timestamp2;

$query_check1="DELETE FROM reservations WHERE name='$name' AND
time_activation_start='$timestamp'";

mysql_query($query_check1, $linkID) or die("Data not found.");

$querytimestamp1="SELECT UNIX_TIMESTAMP('$timestamp')";
$resulttimestamp1=mysql_query($querytimestamp1, $linkID);
$rowtimestamp1 = mysql_fetch_assoc($resulttimestamp1);
$timestamp_start1=$rowtimestamp1["UNIX_TIMESTAMP('$timestamp')"];

$event1=$name."_2".$timestamp_start1;
$query1="DROP EVENT $event1";
mysql_query($query1, $linkID);

$event=$name."_".$timestamp_start1;
$query2 ="DROP EVENT $event";
mysql_query($query2, $linkID);

$responsePayloadString = <<<XML
<cancelreservationsresponse>
<result>You have cancelled the reservation of the slice $name $timestamp</result>
</cancelreservationsresponse>
XML;

$outMessage = new WSMMessage($responsePayloadString);

return $outMessage;
}

$operations = array("cancelreservations"=>"cancelreservations");

```

```

$service = new WSService(array ("operations" => $operations));
// process client requests and reply

$service->reply();
?>

```

reservationcancel.php

```

<?php
session_start();
$owner=$_SESSION['username'];

Echo "<html>";
Echo "<title>List reservations</title>";

$client = new SoapClient("listslices12.wsdl");
$response1 = $client->getListMyReservedSlices($owner);
//$response3 = $client->getListActivatedSlices();
$response2=$response1->validatearray;

if(in_array(2,$response2))
{
echo "not valid rspec";
}
else
{
Echo "<h1>List of Reservations</h1>";
$response=$response1->slicearray;

if (count($response)==0)
{
echo "There is not any reserved slice <br/>";
echo "<a href='\"http://sorch.netmode.ntua.gr/ws/services.php\"'>Return to the services</a><br/>";
}
else{
Echo "<ul>";
Echo "<form method='\"post\"' action='\"http://sorch.netmode.ntua.gr/ws/reservationcancel2.php\"'>";
for($x = 0 ; $x < count($response) ; $x++){
$time_start=$response[$x]->time_start;
$array_start=split("",$time_start);

$time_end=$response[$x]->time_end;
$array_end=split("",$time_end);
$name=$response[$x]->name;
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner;
Echo " has been reserved ";
for($y = 0 ; $y < count($array_start) ; $y++){
if ($y==0)
{
Echo " from ".$array_start[$y]." to ".$array_end[$y];
}
else
{
Echo " and from ".$array_start[$y]." to ".$array_end[$y];
}
}

Echo ".The slice contains the following nodes of testbed: </li>";

$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\",$string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";

```

```

Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];
Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
Echo "</li>";

}
}
Echo "</ul>";
}
Echo "</ol>";

?>
<p><b>Reservations</b></p>
<?php
for($y = 0 ; $y < count($array_start) ; $y++){

$array=split(" ",$array_start[$y]);
$name1=$name.";".$array[0].";".$array[1];
Echo "<input type=\"checkbox\" name=\"name[]\" value=$name1>from ".$array_start[$y]." to
".$array_end[$y];
Echo "<br/>";
}
Echo "<br/>";
Echo "<br/>";
}
Echo "</ul>";
?>
<input type="submit" value="submit" name="getSlice">
<input type="reset" value="Reset" name="reset">

</form>
<?php
}
}
Echo "</html>";
?>

```

reservationcancel2.php

```

<?php
$getSlice = isset($_POST["getSlice"]) ? $_POST['getSlice'] : '';
$name = isset($_POST["name"]) ? $_POST['name'] : '';
?>

<html>
<head>
<title> Cancel Reservations </title>
</head>

```

```

<body>

<?php

if (isset($_POST['getSlice']))
{
    $names=$_POST['name'];
    $string_names='';
    $number_names=count($names);

    for($x = 0 ; $x < $number_names ; $x++){

        $string_names=$string_names.$names[$x];
        if ($x<($number_names-1))
        {
            $string_names=$string_names.',';
        }
    }

if ((count($names)!=0))
    {
    try {

$client = new WSClient(array("to" =>
"http://sorch.netmode.ntua.gr/ws/cancelreservationservice.php", "useSOAP" => 1.1));

$array = split(",", $string_names);
for ($i=0; $i<count($array); $i++){
$name=$array[$i];

$reqPayloadString1 = <<<XML
<ns1:cancelreservations xmlns:ns1="http://sorch.netmode.ntua.gr/ws/cancelreservationservice.php">
<name>$name</name>
</ns1:cancelreservations>
XML;

$reqMessage1 = new WSMMessage($reqPayloadString1);

    $response1 = $client->request($reqMessage1);
    $simplexml1 = new SimpleXMLElement($response1->str);

    echo $simplexml1->result."<br/>";
}

} catch (Exception $e) {

    if ($e instanceof WSFault) {
        printf("Soap Fault: %s\n", $e->Reason);
    } else {
        printf("Message = %s\n", $e->getMessage());
    }
}
else
{
echo "You have not canceled any reservation <br/>";
}
echo "<a href='\"http://sorch.netmode.ntua.gr/ws/services.php\"'>Return to the services</a><br/>";
}
?>

</body>
</html>

```

delsliceservice.php

```

<?php
//delete a slice of testbed
function delslice($message){
$simplexml = new SimpleXMLElement($message->str);
$name =$simplexml->name;
$host = "localhost";
$user = "omf";

```

```

$pass = "omf";
$database = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($database, $linkID) or die("Could not find database.");

$query="SELECT slice_id FROM slices WHERE name='$name'";
$result = mysql_query($query, $linkID) or die("Data not found.");

$row = mysql_fetch_assoc($result);
$slice_id=$row['slice_id'];
if ($row<1)
{
$responsePayloadString = <<<XML
<delsliceresponse>
<result>This slice with name $name does not exist</result>
</delsliceresponse>
XML;

$outMessage = new WSMMessage($responsePayloadString);

return $outMessage;
}
else
{
$query1="DELETE FROM slices WHERE name='$name'";
mysql_query($query1, $linkID);

$query2="DELETE FROM slice_node_association WHERE slice_id='$slice_id'";
mysql_query($query2, $linkID);

$query_check1="SELECT * FROM reservations WHERE name='$name'";

$result_check1 = mysql_query($query_check1, $linkID) or die("Data not found.");
$num_rows=mysql_num_rows($result_check1);

for($x = 0 ; $x < mysql_num_rows($result_check1) ; $x++){
    $row_check1 = mysql_fetch_assoc($result_check1);

    $timestamp_start_check=$row_check1['time_activation_start'];
    $timestamp_end_check=$row_check1['time_activation_end'];

$querytimestamp1="SELECT UNIX_TIMESTAMP('$timestamp_start_check')";
$resulttimestamp1=mysql_query($querytimestamp1, $linkID);
$rowtimestamp1 = mysql_fetch_assoc($resulttimestamp1);
$timestamp_start1=$rowtimestamp1["UNIX_TIMESTAMP('$timestamp_start_check')"];

$querytimestamp2="SELECT UNIX_TIMESTAMP('$timestamp_end_check')";
$resulttimestamp2=mysql_query($querytimestamp2, $linkID);
$rowtimestamp2= mysql_fetch_assoc($resulttimestamp2);
$timestamp_end1=$rowtimestamp2["UNIX_TIMESTAMP('$timestamp_end_check')"];

$event1=$name."_2".$timestamp_start1;

$query5="DROP EVENT $event1";
mysql_query($query5, $linkID);

$event=$name."_".$timestamp_start1;
$query6 ="DROP EVENT $event";
mysql_query($query6, $linkID);
}

$query3="DELETE FROM reservations WHERE name='$name'";
mysql_query($query3, $linkID);

$query4="DELETE FROM node_if_association WHERE slice_id='$slice_id'";
mysql_query($query4, $linkID);

$responsePayloadString = <<<XML
<delsliceresponse>
<result>You deleted the slice $name</result>
</delsliceresponse>
XML;

$outMessage = new WSMMessage($responsePayloadString);

```



```

return $outMessage;
}
}
$operations = array("delslice"=>"delslice");
$service = new WSService(array ("operations" => $operations));
// process client requests and reply
$service->reply();
?>

```

delslice.php

```

<?php
//select the slices you want to delete
session_start();
$owner=$_SESSION['username'];

Echo "<html>";
Echo "<title>Delete slices</title>";

$client = new SoapClient("listslices12.wsdl");
$client1 = new SoapClient("listslices12.wsdl");
$client2 = new SoapClient("listslices12.wsdl");
$response5 = $client->getListMyActivatedSlices($owner);
$response4=$response5->validatearray;
$response3 = $client1->getListMyReservedSlices($owner);
$response1 = $client2->getListMyNoReservedSlices($owner);
$response2=$response1->validatearray;
$response6=$response3->validatearray;

if((in_array(2,$response2)) || (in_array(2,$response4)) || (in_array(2,$response6)))
{
echo "not valid rspec";
}
else
{
$response=$response5->slicearray;
$reserved=array();
for($x = 0 ; $x < count($response) ; $x++){
$reserved[]=$response[$x]->name;
}

Echo "<form method=\"post\" action=\"http://sorch.netmode.ntua.gr/ws/delslice2.php\">";
Echo "<h1>Select slices to delete</h1>";
$response=$response1->slicearray;
Echo "<h3>No-activated slices</h3>";
Echo "<p><b>List of no-reserved Slices which are not activate now</b></p>";
if (count($response)==0)
{
echo "There is not any noreserved slice that is noactivated <br/>";
}
else{
for($x = 0 ; $x < count($response) ; $x++){

$name1=$response[$x]->name;
Echo "<input type=\"checkbox\" name=\"name[]\" value=$name1>The ".$response[$x]->name." with owner
".$response[$x]->owner." contains the following nodes of testbed: ";
$rspec=$response[$x]->rspec;

$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\",$string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;

Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interface1)
{

```

```

if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$checkl=$interfacel['client_id'];
$array=split(":",$checkl);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){

$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";

}
}
Echo "</ul>";
}

Echo "</ol>";
}

}

$response=$response3->slicearray;
Echo "<p><b>List of reserved Slices which are not activate now</b></p>";
if (count($response)==0)
{
echo "There is not any reserved slice that is noactivated now <br/>";
}
else{

for($x = 0 ; $x < count($response) ; $x++){
if(!in_array($response[$x]->name,$reserved)){
$time_start=$response[$x]->time_start;
$array_start=split("",$time_start);

$time_end=$response[$x]->time_end;
$array_end=split("",$time_end);

$name1=$response[$x]->name;
Echo "<input type=\"checkbox\" name=\"name[]\" value=$name1>The ".$response[$x]->name." with owner
".$response[$x]->owner;
Echo " has been reserved ";
for($y = 0 ; $y < count($array_start) ; $y++){

if ($y==0)
{
Echo " from ".$array_start[$y]." to ".$array_end[$y];
}
else
{
Echo " and from ".$array_start[$y]." to ".$array_end[$y];
}
}

Echo ".The slice contains the following nodes of testbed: ";
$rspec=$response[$x]->rspec;

```

```

$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;

Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
}
Echo "</ul>";
}

Echo "</ol>";
}
}

Echo "<h3>Activated Slices</h3>";
$response=$response5->slicearray;

if (count($response)==0)
{
echo "There is not any activated slice <br/>";
}
else{
for($x = 0 ; $x < count($response) ; $x++){
$name1=$response[$x]->name;
Echo "<input type=\"checkbox\" name=\"name[]\" value=$name1>The ".$response[$x]->name." with owner
".$response[$x]->owner." , which has been activated from ".$response[$x]->time_start." and for
".$response[$x]->duration." hours contains the following nodes of testbed: ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

```

```

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;

Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check1=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){

$nodeName=$features->getName();

$value=$entry3->$nodeName;

Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}

}
Echo "</li>";
}
}
Echo "</ul>";
}
Echo "</ol>";
}
?>
<input type="submit" value="submit" name="getSlice">
<input type="reset" value="Reset" name="reset">

</form>
<?php
}
?>

```

delslice2.php

```

<?php
//confirmation that specific slices are deleted
$getSlice = isset($_POST["getSlice"]) ? $_POST['getSlice'] : '';
$name = isset($_POST["name"]) ? $_POST['name'] : '';
?>

<html>
<head>
<title> DELETE A SLICE OF TESTBED </title>
</head>
<body>

```

```

<?php

if (isset($_POST['getSlice']))
{
    $names=$_POST['name'];
    $string_names='';
    $number_names=count($names);

    for($x = 0 ; $x < $number_names ; $x++){

        $string_names=$string_names.$names[$x];
        if ($x<($number_names-1))
        {
            $string_names=$string_names.',';
        }
    }

if ((count($names)!=0))
    {
try {

$client = new WSClient(array("to" =>
"http://sorch.netmode.ntua.gr/ws/delsliceservice.php", "useSOAP" => 1.1));

$array = split(",",$string_names);
for ($i=0; $i<count($array); $i++){
$name=$array[$i];

$reqPayloadString1 = <<<XML
<ns1:delslice xmlns:ns1="http://sorch.netmode.ntua.gr/ws/delsliceservice.php">
<name>$name</name>
</ns1:delslice>
XML;

$reqMessage1 = new WSMMessage($reqPayloadString1);

$response1 = $client->request($reqMessage1);
$simplexml1 = new SimpleXMLElement($response1->str);

echo $simplexml1->result."<br/>";
}

} catch (Exception $e) {

    if ($e instanceof WSFault) {
        printf("Soap Fault: %s\n", $e->Reason);
    } else {
        printf("Message = %s\n",$e->getMessage());
    }
}
else
{
echo "You have not deleted any slice <br/>";
}
echo "<a href='\"http://sorch.netmode.ntua.gr/ws/services.php\"'>Return to the services</a><br/>";
}
?>

</body>
</html>

```

listslicesservicenew2.php

```

<?php

class slice{
public $name;
public $owner;
public $time_start;
public $time_end;
public $duration;
public $rspec;
}

```

```

class testbedslices {
public $slicearray;
public $validatearray;
}

//function for getting the available resources in a specific time duration
function getAvailableResources($time){
$host = "localhost";
$user = "omf";
$password = "omf";
$database = "inventory2";

$linkID = mysql_connect($host, $user, $password) or die("Could not connect to host.");
mysql_select_db($database, $linkID) or die("Could not find database.");

include 'example3.php';
$rates="1 2 5.5 6 9 11 12 18 24 36 48 54";
$modes=array("AP","STA");
$sxe = new SimpleXMLElement($xmlstr);

$array=array();
$array=split(",",$time);

$year=$array[0];
$month=$array[1];
$day=$array[2];

$hour_start=$array[3];
$duration=$array[4];

$date=$year."-".$month."-".$day;

    $hour_init=$hour_start.":00:00";

$timestamp=$date." ".$hour_init;

$timestamp_start=strtotime($timestamp);

$duration_timestamp=3600*$duration;

$timestamp_end=$timestamp_start+$duration_timestamp;

$query_check1="SELECT * FROM reservations";

$result_check1 = mysql_query($query_check1, $linkID) or die("Data not found.");
$num_rows=mysql_num_rows($result_check1);

$names=array();
$nodes=array();

for($x = 0 ; $x < mysql_num_rows($result_check1) ; $x++){
    $row_check1 = mysql_fetch_assoc($result_check1);

        $timestamp_start_check=$row_check1['time_activation_start'];
        $timestamp_end_check=$row_check1['time_activation_end'];

$querytimestamp1="SELECT UNIX_TIMESTAMP('$timestamp_start_check')";
$resulttimestamp1=mysql_query($querytimestamp1, $linkID);
$rowtimestamp1 = mysql_fetch_assoc($resulttimestamp1);
$timestamp_start1=$rowtimestamp1["UNIX_TIMESTAMP('$timestamp_start_check')"];

$querytimestamp2="SELECT UNIX_TIMESTAMP('$timestamp_end_check')";
$resulttimestamp2=mysql_query($querytimestamp2, $linkID);
$rowtimestamp2= mysql_fetch_assoc($resulttimestamp2);
$timestamp_end1=$rowtimestamp2["UNIX_TIMESTAMP('$timestamp_end_check')"];

$check4=($timestamp_start<=$timestamp_start1) && ($timestamp_end<=$timestamp_start1);
$check5=($timestamp_start>=$timestamp_end1) && ($timestamp_end>=$timestamp_end1);

if ($check4 || $check5)
{
$flag=1;
}
else
{
$flag=1;
}
}

```

```

$slice_name=$row_check1['name'];
if (!in_array($slice_name,$names))
{
$names[]=$slice_name;
$query_check2="SELECT slice_id FROM slices WHERE name='$slice_name'";
$result_check2 = mysql_query($query_check2, $linkID) or die("Data not found.");
$row_check2 = mysql_fetch_assoc($result_check2);
$slice_id_check=$row_check2['slice_id'];

$query_check3="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id_check}";
$result_check3 = mysql_query($query_check3, $linkID) or die("Data not found.");
$num_rows2=mysql_num_rows($result_check3);

for($y = 0 ; $y < mysql_num_rows($result_check3) ; $y++){
    $row_check3= mysql_fetch_assoc($result_check3);
    $node_id_check=$row_check3['node_id'];
if(!in_array($node_id_check,$nodes))
{
    $nodes[]=$node_id_check;
}
}
}

$query_check1="SELECT * FROM reservations";

$result_check1 = mysql_query($query_check1, $linkID) or die("Data not found.");
$num_rows=mysql_num_rows($result_check1);

$channels=array();
$names1=array();

for($x = 0 ; $x < mysql_num_rows($result_check1) ; $x++){
    $row_check1 = mysql_fetch_assoc($result_check1);

    $timestamp_start_check=$row_check1['time_activation_start'];
    $timestamp_end_check=$row_check1['time_activation_end'];

    $querytimestamp1="SELECT UNIX_TIMESTAMP('$timestamp_start_check')";
    $resulttimestamp1=mysql_query($querytimestamp1, $linkID);
    $rowtimestamp1 = mysql_fetch_assoc($resulttimestamp1);
    $timestamp_start1=$rowtimestamp1["UNIX_TIMESTAMP('$timestamp_start_check')"];

    $querytimestamp2="SELECT UNIX_TIMESTAMP('$timestamp_end_check')";
    $resulttimestamp2=mysql_query($querytimestamp2, $linkID);
    $rowtimestamp2= mysql_fetch_assoc($resulttimestamp2);
    $timestamp_end1=$rowtimestamp2["UNIX_TIMESTAMP('$timestamp_end_check')"];

    $check4=($timestamp_start<=$timestamp_start1) && ($timestamp_end<=$timestamp_start1);
    $check5=($timestamp_start>=$timestamp_end1) && ($timestamp_end>=$timestamp_end1);

if ($check4 || $check5)
{
    $flag=1;
}
else
{
    $slice_name=$row_check1['name'];
if (!in_array($slice_name,$names1))
{
    $names1[]=$slice_name;
    $query_check2="SELECT slice_id FROM slices WHERE name='$slice_name'";
    $result_check2 = mysql_query($query_check2, $linkID) or die("Data not found.");
    $row_check2 = mysql_fetch_assoc($result_check2);
    $slice_id_check=$row_check2['slice_id'];

    $query_check3="SELECT RadioChannel FROM node_if_association WHERE slice_id={$slice_id_check}";
    $result_check3 = mysql_query($query_check3, $linkID) or die("Data not found.");
    $num_rows2=mysql_num_rows($result_check3);

for($y = 0 ; $y < mysql_num_rows($result_check3) ; $y++){
    $row_check3= mysql_fetch_assoc($result_check3);
    $RadioChannel=$row_check3['RadioChannel'];
if(!in_array($RadioChannel,$channels))
{

```

```

$channels[]=$RadioChannel;
}
}
}
}
if (count($channels)==0)
{
$channels[]=0;
}

$channels_av=array();
for ($y=0;$y<13;$y++)
{
$temp=$y+1;
if (!in_array($temp,$channels))
{
$channels_av[]=$temp;
}
}

if ($flag==1){
$query= "SELECT * FROM nodes";
$result = mysql_query($query, $linkID) or die("Data not found.");

for ($x = 0 ; $x < mysql_num_rows($result) ; $x++){
    $check=0;
    $row = mysql_fetch_assoc($result);
    $id_check_init=$row['id'];
    for ($y = 0 ; $y < count($nodes) ; $y++){
        if ($nodes[$y]==$id_check_init)
        {
            $check=1;
        }
    }
    if ($check==0)
    {

        $node=$sxe->addChild('node');

        $id=$row['id'];
        $node->addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+node+node".$id);
        $node-
>addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
        $node->addAttribute('component_name',"node".$id);
        $node->addAttribute('exclusive','true');
        $available=$node->addChild('available');
        $available->addAttribute('now','true');

        $hardware=$node->addChild('hardware_type');

        $hardware->addAttribute('name','wireless access point');

        $sliver=$node->addChild('sliver_type');

        $name1="node".$id;
        $sliver->addAttribute('name','access point');

        for ($y=0;$y<2;$y++){
            $interface=$node->addChild('interface');

            $if=$name1."if".$y;
            $interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
            $interface->addAttribute('component_name',$if);
            $wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");

            $child=$wireless-
>addChild('Rate',$rates,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");

            for ($z=0;$z<count($channels_av);$z++){

```



```

        $child=$wireless-
>addChild('RadioChannel',$channels_av[$z],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if")
;
    }
    for ($z=0;$z<count($modes);$z++){
        $child=$wireless-
>addChild('OperatingMode',$modes[$z],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
    }
}
}
}
}
}
$string1=$sxe->asXML();

$xmlDoc = new DomDocument();
$xmlSchema = '/var/www/ws/RSpec/ad.xsd';
//Load the xml document in the DOMDocument object
$xmlDoc->loadXML($string1);

if ($xmlDoc->schemaValidate($xmlSchema)) {
return $string1;
}
else
{
return "not valid";
}
}
}
}

//function for getting the resources allocated in a slice
function getListSliceResources($name){

$host = "localhost";
$user = "omf";
$pass = "omf";
$database = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($database, $linkID) or die("Could not find database.");

include 'manifest.php';

$sxe = new SimpleXMLElement($xmlstr);
$query_init="SELECT slice_id FROM slices WHERE name='$name'";
$result_init = mysql_query($query_init, $linkID) or die("Data not found.");

$row_init = mysql_fetch_assoc($result_init);
$slice_id_check=$row_init['slice_id'];

$query1="SELECT * FROM slice_node_association WHERE slice_id='$slice_id_check'";
$result1 = mysql_query($query1, $linkID) or die("Data not found.");

$testbed=array();
for($x = 0 ; $x < mysql_num_rows($result1) ; $x++){
    $row1 = mysql_fetch_assoc($result1);
    $node_id=$row1['node_id'];
    $query= "SELECT * FROM nodes WHERE id={$node_id}";
    $result = mysql_query($query, $linkID) or die("Data not found.");
    $row= mysql_fetch_assoc($result);

    $node=$sxe->addChild('node');

    $id=$row['id'];
    $node->addAttribute('client_id',"node".$id);
    $node->addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+node+node".$id);
    $node-
>addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
    $node->addAttribute('sliver_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+sliver+node".$id);
    $node->addAttribute('exclusive','true');

    $hardware=$node->addChild('hardware_type');

$hardware->addAttribute('name','wireless access point');

$sliver=$node->addChild('sliver_type');

```

```

$name1="node".$id;
$sliver->addAttribute('name','access point');

$query2= "SELECT * FROM node_if_association WHERE slice_id={$slice_id_check} AND
node_id={$node_id}";
$result2 = mysql_query($query2, $linkID) or die("Data not found.");

for($y = 0 ; $y < mysql_num_rows($result2) ; $y++){
    $interface=$node->addChild('interface');

    $row2 = mysql_fetch_assoc($result2);
    $if1=$row2['if_id'];
    $if=$name1.".".$if1;
    $interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
    $interface->addAttribute('client_id',$if);

    $wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
    $array=array_keys($row2);

    for ($z=3;$z<count($array);$z++){
        $child=$wireless-
>addChild($array[$z],$row2[$array[$z]],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
    }
}
}
}
$string1=$sxe->asXML();

$xmlDoc = new DomDocument();
$xmlSchema = '/var/www/ws/RSpec/manifest.xsd';
//Load the xml document in the DOMDocument object
$xmlDoc->loadXML($string1);

if ($xmlDoc->schemaValidate($xmlSchema)) {
    return $string1;
}
else
{
    return "not valid";
}
}

//function for getting the available resources for a specific slice
function getListAvailableResources($name){

    $host = "localhost";
    $user = "omf";
    $pass = "omf";
    $database = "inventory2";

    $linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
    mysql_select_db($database, $linkID) or die("Could not find database.");

    include 'example3.php';

    $rates="1 2 5.5 6 9 11 12 18 24 36 48 54";
    $modes=array("AP","STA");

    $sxe = new SimpleXMLElement($xmlstr);

    $query="SELECT * FROM reservations WHERE name='$name'";
    $result = mysql_query($query, $linkID) or die("Data not found.");

    $array_start=array();
    $array_end=array();

    for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
        $row = mysql_fetch_assoc($result);

```

```

$name=$row['name'];
$timestamp_end_check=$row['time_activation_end'];

$querytimestamp2="SELECT UNIX_TIMESTAMP('$timestamp_end_check')";
$resulttimestamp2=mysql_query($querytimestamp2, $linkID);
$rowtimestamp2= mysql_fetch_assoc($resulttimestamp2);
$timestamp_end1=$rowtimestamp2["UNIX_TIMESTAMP('$timestamp_end_check')"];

$timestamp_start_check=$row['time_activation_start'];

$querytimestamp3="SELECT UNIX_TIMESTAMP('$timestamp_start_check')";
$resulttimestamp3=mysql_query($querytimestamp3, $linkID);
$rowtimestamp3= mysql_fetch_assoc($resulttimestamp3);
$timestamp_start1=$rowtimestamp3["UNIX_TIMESTAMP('$timestamp_start_check')"];

$time=time();
if (($timestamp_end1>$time))
{
    $array_start[]=$timestamp_start1;
    $array_end[]=$timestamp_end1;
}
}

if (count($array_start)==0)
{
    $array_start[]=0;
    $array_end[]=0;
}

$query_check1="SELECT * FROM reservations";

$result_check1 = mysql_query($query_check1, $linkID) or die("Data not found.");
$num_rows=mysql_num_rows($result_check1);

$names=array();
$nodes=array();

for($x = 0 ; $x < mysql_num_rows($result_check1) ; $x++){
    $row_check1 = mysql_fetch_assoc($result_check1);

    $timestamp_start_check=$row_check1['time_activation_start'];
    $timestamp_end_check=$row_check1['time_activation_end'];

    $querytimestamp1="SELECT UNIX_TIMESTAMP('$timestamp_start_check')";
    $resulttimestamp1=mysql_query($querytimestamp1, $linkID);
    $rowtimestamp1 = mysql_fetch_assoc($resulttimestamp1);
    $timestamp_start1=$rowtimestamp1["UNIX_TIMESTAMP('$timestamp_start_check')"];

    $querytimestamp2="SELECT UNIX_TIMESTAMP('$timestamp_end_check')";
    $resulttimestamp2=mysql_query($querytimestamp2, $linkID);
    $rowtimestamp2= mysql_fetch_assoc($resulttimestamp2);
    $timestamp_end1=$rowtimestamp2["UNIX_TIMESTAMP('$timestamp_end_check')"];

    $time=time();
    for ($z=0;$z<count($array_start);$z++)
    {
        $timestamp_start=$array_start[$z];
        $timestamp_end=$array_end[$z];
        $time=time();
        $check4=($timestamp_start<=$timestamp_start1) && ($timestamp_end<=$timestamp_start1) ;
        $check5=($timestamp_start>=$timestamp_end1) && ($timestamp_end>=$timestamp_end1) ;

        if ($check4 || $check5)
        {
            $flag=1;
        }
        else
        {
            $flag=1;
            $slice_name=$row_check1['name'];
            if (!in_array($slice_name,$names))
            {
                $names[]=$slice_name;
                $query_check2="SELECT slice_id FROM slices WHERE name='$slice_name'";
            }
        }
    }
}

```

```

$result_check2 = mysql_query($query_check2, $linkID) or die("Data not found.");
$row_check2 = mysql_fetch_assoc($result_check2);
$slice_id_check=$row_check2['slice_id'];

$query_check3="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id_check}";
$result_check3 = mysql_query($query_check3, $linkID) or die("Data not found.");
$num_rows2=mysql_num_rows($result_check3);

for($y = 0 ; $y < mysql_num_rows($result_check3) ; $y++){
    $row_check3= mysql_fetch_assoc($result_check3);
    $node_id_check=$row_check3['node_id'];
    if(!in_array($node_id_check,$nodes))
    {
        $nodes[]=$node_id_check;
    }
}

$query="SELECT * FROM reservations WHERE name='$name'";
$result = mysql_query($query, $linkID) or die("Data not found.");

$array_start1=array();
$array_end1=array();

for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
    $row = mysql_fetch_assoc($result);

    $name=$row['name'];
    $timestamp_end_check=$row['time_activation_end'];

    $querytimestamp2="SELECT UNIX_TIMESTAMP('$timestamp_end_check')";
    $resulttimestamp2=mysql_query($querytimestamp2, $linkID);
    $rowtimestamp2= mysql_fetch_assoc($resulttimestamp2);
    $timestamp_end1=$rowtimestamp2["UNIX_TIMESTAMP('$timestamp_end_check')"];

    $timestamp_start_check=$row['time_activation_start'];

    $querytimestamp3="SELECT UNIX_TIMESTAMP('$timestamp_start_check')";
    $resulttimestamp3=mysql_query($querytimestamp3, $linkID);
    $rowtimestamp3= mysql_fetch_assoc($resulttimestamp3);
    $timestamp_start1=$rowtimestamp3["UNIX_TIMESTAMP('$timestamp_start_check')"];

    $time=time();
    if (($timestamp_end1>$time))
    {
        $array_start1[]=$timestamp_start1;
        $array_end1[]=$timestamp_end1;
    }
}

if (count($array_start1)==0)
{
    $array_start1[]=0;
    $array_end1[]=0;
}

$query_check1="SELECT * FROM reservations WHERE name<>'$name'";

$result_check1 = mysql_query($query_check1, $linkID) or die("Data not found.");
$num_rows=mysql_num_rows($result_check1);

$names1=array();
$channels=array();

for($x = 0 ; $x < mysql_num_rows($result_check1) ; $x++){
    $row_check1 = mysql_fetch_assoc($result_check1);

    $timestamp_start_check=$row_check1['time_activation_start'];
    $timestamp_end_check=$row_check1['time_activation_end'];

    $querytimestamp1="SELECT UNIX_TIMESTAMP('$timestamp_start_check')";

```

```

$resulttimestamp1=mysql_query($querytimestamp1, $linkID);
$rowtimestamp1 = mysql_fetch_assoc($resulttimestamp1);
$timestamp_start1=$rowtimestamp1["UNIX_TIMESTAMP('timestamp_start_check')"];

$querytimestamp2="SELECT UNIX_TIMESTAMP('$timestamp_end_check')";
$resulttimestamp2=mysql_query($querytimestamp2, $linkID);
$rowtimestamp2= mysql_fetch_assoc($resulttimestamp2);
$timestamp_end1=$rowtimestamp2["UNIX_TIMESTAMP('timestamp_end_check')"];

$time=time();
for ($z=0;$z<count($array_start1);$z++)
{
$timestamp_start=$array_start1[$z];
$timestamp_end=$array_end1[$z];

$check4=($timestamp_start<=$timestamp_start1) && ($timestamp_end<=$timestamp_start1);
$check5=($timestamp_start>=$timestamp_end1) && ($timestamp_end>=$timestamp_end1);

if ($check4 || $check5)
{
$flag=1;
}
else
{
$flag=1;
$slice_name=$row_check1['name'];
if (!in_array($slice_name,$names1))
{
$names1[]=$slice_name;
$query_check2="SELECT slice_id FROM slices WHERE name='$slice_name'";
$result_check2 = mysql_query($query_check2, $linkID) or die("Data not found.");
$row_check2 = mysql_fetch_assoc($result_check2);
$slice_id_check=$row_check2['slice_id'];

$query_check3="SELECT RadioChannel FROM node_if_association WHERE slice_id={$slice_id_check}";
$result_check3 = mysql_query($query_check3, $linkID) or die("Data not found.");
$num_rows2=mysql_num_rows($result_check3);

for($y = 0 ; $y < mysql_num_rows($result_check3) ; $y++){
    $row_check3= mysql_fetch_assoc($result_check3);
    $RadioChannel=$row_check3['RadioChannel'];
    if(!in_array($RadioChannel,$channels))
    {
$channels[]=$RadioChannel;
}
}
}
}
}

$channels_av=array();
for($y=0;$y<13;$y++)
{
$temp=$y+1;
if(!in_array($temp,$channels))
{
$channels_av[]=$temp;
}
}

if ($flag==1){
$query= "SELECT * FROM nodes";
$result = mysql_query($query, $linkID) or die("Data not found.");

for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
    $check=0;
    $row = mysql_fetch_assoc($result);
    $id_check_init=$row['id'];
    for($y = 0 ; $y < count($nodes) ; $y++){
        if ($nodes[$y]==$id_check_init)
        {
            $check=1;
        }
    }
    if ($check==0)

```

```

{

$node=$sxe->addChild('node');

$id=$row['id'];
$node->addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+node+node".$id);
$node-
>addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
$node->addAttribute('component_name',"node".$id);
$node->addAttribute('exclusive','true');
$available=$node->addChild('available');
$available->addAttribute('now','true');

$hardware=$node->addChild('hardware_type');

$hardware->addAttribute('name','wireless access point');

$sliver=$node->addChild('sliver_type');

$name1="node".$id;
$sliver->addAttribute('name','access point');

for ($y=0;$y<2;$y++){
$interface=$node->addChild('interface');

$if=$name1.":if".$y;
$interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
$interface->addAttribute('component_name',$if);
$wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");

$child=$wireless-
>addChild('Rate',$rates,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");

for ($z=0;$z<count($channels_av);$z++){
$child=$wireless-
>addChild('RadioChannel',$channels_av[$z],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if")
;
}
for ($z=0;$z<count($modes);$z++){
$child=$wireless-
>addChild('OperatingMode',$modes[$z],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
}
}
}

$query_check="SELECT slice_id FROM slices WHERE name='$name'";
$result_check = mysql_query($query_check, $linkID) or die("Data not found.");
$row_check= mysql_fetch_assoc($result_check);
$slice_id=$row_check['slice_id'];

$query= "SELECT * FROM slice_node_association WHERE slice_id={$slice_id}";
$result = mysql_query($query, $linkID) or die("Data not found.");

for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
$row = mysql_fetch_assoc($result);

$node=$sxe->addChild('node');

$id=$row['node_id'];
$node->addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+node+node".$id);
$node-
>addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
$node->addAttribute('component_name',"node".$id);
$node->addAttribute('exclusive','true');
$available=$node->addChild('available');
$available->addAttribute('now','true');

$hardware=$node->addChild('hardware_type');

$hardware->addAttribute('name','pc');

```

```

    $sliver=$node->addChild('sliver_type');

    $name1="node".$id;
    $sliver->addAttribute('name',$name1);

    for ($y=0;$y<2;$y++){
        $interface=$node->addChild('interface');

        $if=$name1.".if".$y;
        $interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
        $interface->addAttribute('component_name',$if);
        $wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");

        for ($z=0;$z<count($rates);$z++){
            $child=$wireless-
>addChild('Rate',$rates[$z],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        }
        for ($z=0;$z<count($channels_av);$z++){
            $child=$wireless-
>addChild('RadioChannel',$channels_av[$z],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        }
        for ($z=0;$z<count($modes);$z++){
            $child=$wireless-
>addChild('OperatingMode',$modes[$z],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        }
    }

    $string1=$sxe->asXML();

    $xdoc = new DomDocument();
    $xmlschema = '/var/www/ws/RSpec/ad.xsd';
    //Load the xml document in the DOMDocument object
    $xdoc->loadXML($string1);

    if ($xdoc->schemaValidate($xmlschema)) {
        return $string1;
    }
    else
    {
        return "not valid";
    }
}

//function for getting a list of slices
function getListSlices(){

    $host = "localhost";
    $user = "omf";
    $pass = "omf";
    $database = "inventory2";

    $linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
    mysql_select_db($database, $linkID) or die("Could not find database.");

    $query="SELECT * FROM slices";
    $result = mysql_query($query, $linkID) or die("Data not found.");

    $testbed= new testbedslices;
    $slicesarray=array();
    $validatearray=array();
    for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
        $row = mysql_fetch_assoc($result);
        $slice= new slice;
        $slice->name=$row['name'];
        $slice->owner=$row['owner'];
        $slice_id=$row['slice_id'];
        $query1="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id}";
        $result1 = mysql_query($query1, $linkID) or die("Data not found.");
        $nodesarray=array();
        include 'manifest.php';
    }
}

```

```

$sxe = new SimpleXMLElement($xmlstr);
for($i = 0 ; $i < mysql_num_rows($result1) ; $i++){
    $row1 = mysql_fetch_assoc($result1);
    $node_id=$row1['node_id'];

    $query2="SELECT * FROM nodes WHERE id={$node_id}";
    $result2= mysql_query($query2, $linkID) or die("Data not found.");
    $row2 = mysql_fetch_assoc($result2);
    $node=$sxe->addChild('node');

    $id=$row2['id'];
    $node->addAttribute('client_id',"node".$id);
    $node->addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+node+node".$id);
    $node-
>addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
    $node->addAttribute('sliver_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+sliver+node".$id);
    $node->addAttribute('exclusive','true');

    $hardware=$node->addChild('hardware_type');

    $hardware->addAttribute('name','wireless access point');

    $sliver=$node->addChild('sliver_type');

    $name1="node".$id;
    $sliver->addAttribute('name','access point');;

    $query3= "SELECT * FROM node_if_association WHERE slice_id={$slice_id} AND
node_id={$node_id}";
    $result3 = mysql_query($query3, $linkID) or die("Data not found.");

    for($y = 0 ; $y < mysql_num_rows($result3) ; $y++){
        $interface=$node->addChild('interface');

        $row3 = mysql_fetch_assoc($result3);
        $if1=$row3['if_id'];
        $if=$name1.".".$if1;
        $interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
        $interface->addAttribute('client_id',$if);
        $wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        $array=array_keys($row3);

        for ($z=3;$z<count($array);$z++){
            $child=$wireless-
>addChild($array[$z],$row3[$array[$z]],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        }
    }
}
$string1=$sxe->asXML();

$slice->rspec=$string1;
$slicesarray[$x]=$slice;

$xmlschema = '/var/www/ws/RSpec/manifest.xsd';
//Load the xml document in the DOMDocument object
$xml->loadXML($string1);

if ($xml->schemaValidate($xmlschema)) {
    $validatearray[$x]=1;
}
else
{
    $validatearray[$x]=2;
}
}
}
$testbed->slicesarray=$slicesarray;
$testbed->validatearray=$validatearray;
return $testbed;

```



```

}

//list the slices of a specific owner
function getListMySlices($owner){

$host = "localhost";
$user = "omf";
$pass = "omf";
$database = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($database, $linkID) or die("Could not find database.");

$query="SELECT * FROM slices WHERE owner='$owner'";
$result = mysql_query($query, $linkID) or die("Data not found.");

$testbed= new testbedslices;
$slicesarray=array();
$validatearray=array();
for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
    $row = mysql_fetch_assoc($result);
    $slice= new slice;
    $slice->name=$row['name'];
    $slice->owner=$row['owner'];
    $slice_id=$row['slice_id'];

    $query1="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id}";
    $result1 = mysql_query($query1, $linkID) or die("Data not found.");

    $nodesarray=array();
    include 'manifest.php';

    $sxe = new SimpleXMLElement($xmlstr);
    for($i = 0 ; $i < mysql_num_rows($result1) ; $i++){
        $row1 = mysql_fetch_assoc($result1);
        $node_id=$row1['node_id'];

        $query2="SELECT * FROM nodes WHERE id={$node_id}";
        $result2= mysql_query($query2, $linkID) or die("Data not found.");
        $row2 = mysql_fetch_assoc($result2);
        $node=$sxe->addChild('node');

        $id=$row2['id'];
        $node->addAttribute('client_id',"node".$id);
        $node->addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+node+node".$id);
        $node-
>addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
        $node->addAttribute('sliver_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+sliver+node".$id);
        $node->addAttribute('exclusive','true');

        $hardware=$node->addChild('hardware_type');

        $hardware->addAttribute('name','wireless access point');

        $sliver=$node->addChild('sliver_type');

        $name1="node".$id;
        $sliver->addAttribute('name','access point');

        $query3= "SELECT * FROM node_if_association WHERE slice_id={$slice_id} AND
node_id={$node_id}";
        $result3 = mysql_query($query3, $linkID) or die("Data not found.");

        for($y = 0 ; $y < mysql_num_rows($result3) ; $y++){
            $interface=$node->addChild('interface');

            $row3 = mysql_fetch_assoc($result3);
            $if1=$row3['if_id'];
            $if=$name1.".".$if1;
            $interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
            $interface->addAttribute('client_id',$if);
            $wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
            $array=array_keys($row3);

```



```

$slice_id=$row['slice_id'];
$query1="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id}";
$result1 = mysql_query($query1, $linkID) or die("Data not found.");

include 'manifest.php';
$sxe = new SimpleXMLElement($xmlstr);
for($i = 0 ; $i < mysql_num_rows($result1) ; $i++){
    $row1 = mysql_fetch_assoc($result1);
    $node_id=$row1['node_id'];

    $query2="SELECT * FROM nodes WHERE id={$node_id}";
    $result2= mysql_query($query2, $linkID) or die("Data not found.");
    $row2 = mysql_fetch_assoc($result2);
    $node=$sxe->addChild('node');

    $id=$row2['id'];
    $node->addAttribute('client_id',"node".$id);
    $node->addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+node+node".$id);
    $node-
>addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
    $node->addAttribute('sliver_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+sliver+node".$id);
    $node->addAttribute('exclusive','true');

    $hardware=$node->addChild('hardware_type');

    $hardware->addAttribute('name','wireless access point');

    $sliver=$node->addChild('sliver_type');

    $name1="node".$id;
    $sliver->addAttribute('name','access point');

    $query3= "SELECT * FROM node_if_association WHERE slice_id={$slice_id} AND
node_id={$node_id}";
    $result3 = mysql_query($query3, $linkID) or die("Data not found.");

    for($y = 0 ; $y < mysql_num_rows($result3) ; $y++){
        $interface=$node->addChild('interface');

        $row3 = mysql_fetch_assoc($result3);
        $if1=$row3['if_id'];
        $if=$name1.".".$if1;
        $interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
        $interface->addAttribute('client_id',$if);
        $wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        $array=array_keys($row3);

        for ($z=3;$z<count($array);$z++){
            $child=$wireless-
>addChild($array[$z],$row3[$array[$z]],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        }
    }
}
$string1=$sxe->asXML();

$slice->rspec=$string1;
$slicesarray[$x]=$slice;

$xmlschema = '/var/www/ws/RSpec/manifest.xsd';
//Load the xml document in the DOMDocument object
$xml->loadXML($string1);

if ($xml->schemaValidate($xmlschema)) {
    $validatearray[$x]=1;
}
else
{

```

```

$validatearray[$x]=2;
}
}
}
$testbed->slicearray=$slicesarray;
$testbed->validatearray=$validatearray;
return $testbed;
}

//list no-reserved slices of a specific user
function getListMyNoReservedSlices($owner){

$host = "localhost";
$user = "omf";
$pass = "omf";
$databse = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($databse, $linkID) or die("Could not find database.");

$names=array();

$query_init="SELECT * FROM reservations";
$result_init = mysql_query($query_init, $linkID) or die("Data not found.");
for($x = 0 ; $x < mysql_num_rows($result_init) ; $x++){
    $row_init = mysql_fetch_assoc($result_init);
    $timestamp=$row_init['time_activation_end'];

$querytimestamp="SELECT UNIX_TIMESTAMP('$timestamp')";
$resulttimestamp=mysql_query($querytimestamp, $linkID);
$rowtimestamp = mysql_fetch_assoc($resulttimestamp);
$timestamp_end=$rowtimestamp["UNIX_TIMESTAMP('$timestamp')"];

$time=time();
if ($time<$timestamp_end)
{
$names[]=$row_init['name'];
}
}

$query="SELECT * FROM slices WHERE owner='$owner'";
$result = mysql_query($query, $linkID) or die("Data not found.");

$testbed= new testbedslices;
$slicesarray=array();
$validatearray=array();
for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
    $row = mysql_fetch_assoc($result);
    $slice= new slice;

    $name=$row['name'];
    if (!in_array($name,$names))
    {
        $slice->name=$row['name'];
        $slice->owner=$row['owner'];
        $slice_id=$row['slice_id'];
        $query1="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id}";
        $result1 = mysql_query($query1, $linkID) or die("Data not found.");
        include 'manifest.php';

        $sxe = new SimpleXMLElement($xmlstr);
        for($i = 0 ; $i < mysql_num_rows($result1) ; $i++){
            $row1 = mysql_fetch_assoc($result1);
            $node_id=$row1['node_id'];

            $query2="SELECT * FROM nodes WHERE id={$node_id}";
            $result2= mysql_query($query2, $linkID) or die("Data not found.");
            $row2 = mysql_fetch_assoc($result2);
            $node=$sxe->addChild('node');

            $id=$row2['id'];
            $node->addAttribute('client_id',"node".$id);
            $node->addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+node+node".$id);
            $node->
            >addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
            $node->addAttribute('sliver_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+sliver+node".$id);

```

```

        $node->addAttribute('exclusive','true');

$hardware=$node->addChild('hardware_type');

$hardware->addAttribute('name','wireless access point');

$sliver=$node->addChild('sliver_type');

$name1="node".$id;
$sliver->addAttribute('name','access point');

        $query3= "SELECT * FROM node_if_association WHERE slice_id={ $slice_id } AND
node_id={ $node_id }";
        $result3 = mysql_query($query3, $linkID) or die("Data not found.");

        for($y = 0 ; $y < mysql_num_rows($result3) ; $y++){
            $interface=$node->addChild('interface');

            $row3 = mysql_fetch_assoc($result3);
            $if1=$row3['if_id'];
            $if=$name1.".".$if1;
            $interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
            $interface->addAttribute('client_id',$if);
            $wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
            $array=array_keys($row3);

            for ($z=3;$z<count($array);$z++){
                $child=$wireless-
>addChild($array[$z],$row3[$array[$z]],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
            }
        }
    }
}
$string1=$sxe->asXML();

$slice->rspec=$string1;
$slicesarray[$x]=$slice;

$xmlschema = new DomDocument();
$xmlschema = '/var/www/ws/RSpec/manifest.xsd';
//Load the xml document in the DOMDocument object
$xmlschema->loadXML($string1);

if ($xmlschema->schemaValidate($xmlschema)) {
    $validatearray[$x]=1;
}
else
{
    $validatearray[$x]=2;
}
}
}
}
$testbed->slicesarray=$slicesarray;
$testbed->validatearray=$validatearray;
return $testbed;
}

//list reserved slices
function getListReservedSlices(){

$host = "localhost";
$user = "omf";
$pass = "omf";
$database = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($database, $linkID) or die("Could not find database.");

$query="SELECT * FROM reservations";

```

```

$result = mysql_query($query, $linkID) or die("Data not found.");

$testbed= new testbedslices;
$slicesarray=array();
$validatearray=array();

$names=array();
$y=0;
for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
    $row = mysql_fetch_assoc($result);

    $name=$row['name'];
    $timestamp_end_check=$row['time_activation_end'];

    $querytimestamp2="SELECT UNIX_TIMESTAMP('$timestamp_end_check')";
    $resulttimestamp2=mysql_query($querytimestamp2, $linkID);
    $rowtimestamp2= mysql_fetch_assoc($resulttimestamp2);
    $timestamp_end1=$rowtimestamp2["UNIX_TIMESTAMP('$timestamp_end_check')"];

    $timestamp_start_check=$row['time_activation_start'];

    $querytimestamp3="SELECT UNIX_TIMESTAMP('$timestamp_start_check')";
    $resulttimestamp3=mysql_query($querytimestamp3, $linkID);
    $rowtimestamp3= mysql_fetch_assoc($resulttimestamp3);
    $timestamp_start1=$rowtimestamp3["UNIX_TIMESTAMP('$timestamp_start_check')"];

    $time=time();
    if ((in_array($name,$names)) && ($timestamp_end1>$time) && ($timestamp_start1>$time))
    {
        $time_start=",$timestamp_start_check;
        $time_end=",$timestamp_end_check;

        $slice->time_start=$time_start;
        $slice->time_end=$time_end;
    }

    if (!(in_array($name,$names)) && ($timestamp_end1>$time) && ($timestamp_start1>$time))
    {
        $slice= new slice;
        $time_start="";
        $time_end="";
        $names[$y]=$name;

        $query_final="SELECT * FROM slices WHERE name='$name'";
        $result_final = mysql_query($query_final, $linkID) or die("Data not found.");
        $row_final = mysql_fetch_assoc($result_final);
        $slice->name=$row_final['name'];
        $slice->owner=$row_final['owner'];

        $time_start=$timestamp_start_check;
        $time_end=$timestamp_end_check;

        $slice->time_start=$time_start;
        $slice->time_end=$time_end;

        $slice_id=$row_final['slice_id'];

        $query1="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id}";
        $result1 = mysql_query($query1, $linkID) or die("Data not found.");

        include 'manifest.php';

        $sxe = new SimpleXMLElement($xmlstr);
        for($i = 0 ; $i < mysql_num_rows($result1) ; $i++){
            $row1 = mysql_fetch_assoc($result1);
            $node_id=$row1['node_id'];

            $query2="SELECT * FROM nodes WHERE id={$node_id}";
            $result2= mysql_query($query2, $linkID) or die("Data not found.");
            $row2 = mysql_fetch_assoc($result2);
            $node=$sxe->addChild('node');

            $id=$row2['id'];
            $node->addAttribute('client_id',"node".$id);
            $node->addAttribute('component_id',"urn:publicid:IDN+sorch.netnode.ntua.gr+node+node".$id);

```

```

    $node-
>addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
    $node->addAttribute('sliver_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+sliver+node".$id);
    $node->addAttribute('exclusive','true');

$hardware=$node->addChild('hardware_type');

$hardware->addAttribute('name','wireless access point');

$sliver=$node->addChild('sliver_type');

$name1="node".$id;
$sliver->addAttribute('name','access point');

    $query3= "SELECT * FROM node_if_association WHERE slice_id={$slice_id} AND
node_id={$node_id}";
    $result3 = mysql_query($query3, $linkID) or die("Data not found.");

    for($y = 0 ; $y < mysql_num_rows($result3) ; $y++){
        $interface=$node->addChild('interface');

        $row3 = mysql_fetch_assoc($result3);
        $if1=$row3['if_id'];
        $if=$name1.".$if1";
        $interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
        $interface->addAttribute('client_id',$if);
        $wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        $array=array_keys($row3);

        for ($z=3;$z<count($array);$z++){
            $child=$wireless-
>addChild($array[$z],$row3[$array[$z]],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        }
    }
}
}
}
$string1=$sxe->asXML();

$slice->rspec=$string1;
$slicesarray[$x]=$slice;

$xdoc = new DomDocument();
$xmlschema = '/var/www/ws/RSpec/manifest.xsd';
//Load the xml document in the DOMDocument object
$xdoc->loadXML($string1);

if ($xdoc->schemaValidate($xmlschema)) {
    $validatearray[$x]=1;
}
else
{
    $validatearray[$x]=2;
}
}
}
}
$testbed->slicearray=$slicesarray;
$testbed->validatearray=$validatearray;
return $testbed;
}

//list reserved slices of a user
function getListMyReservedSlices($owner){

$host = "localhost";
$user = "omf";
$pass = "omf";
$database = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");

```

```

mysql_select_db($database, $linkID) or die("Could not find database.");

$query="SELECT * FROM reservations";
$result = mysql_query($query, $linkID) or die("Data not found.");

$testbed= new testbedslices;
$slicesarray=array();
$validatearray=array();

$names=array();
$y=0;
for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
    $row = mysql_fetch_assoc($result);
    $name=$row['name'];
    $timestamp_end_check=$row['time_activation_end'];

    $querytimestamp2="SELECT UNIX_TIMESTAMP('$timestamp_end_check')";
    $resulttimestamp2=mysql_query($querytimestamp2, $linkID);
    $rowtimestamp2= mysql_fetch_assoc($resulttimestamp2);
    $timestamp_end1=$rowtimestamp2["UNIX_TIMESTAMP('$timestamp_end_check')"];

    $timestamp_start_check=$row['time_activation_start'];

    $querytimestamp3="SELECT UNIX_TIMESTAMP('$timestamp_start_check')";
    $resulttimestamp3=mysql_query($querytimestamp3, $linkID);
    $rowtimestamp3= mysql_fetch_assoc($resulttimestamp3);
    $timestamp_start1=$rowtimestamp3["UNIX_TIMESTAMP('$timestamp_start_check')"];

    $time=time();
    if ((in_array($name,$names)) && ($timestamp_end1>$time) && ($timestamp_start1>$time))
    {
        $time_start=",$timestamp_start_check;
        $time_end=",$timestamp_end_check;

        $slice->time_start=$time_start;
        $slice->time_end=$time_end;
    }

    if ((!in_array($name,$names)) && ($timestamp_end1>$time) && ($timestamp_start1>$time))
    {
        $slice= new slice;
        $time_start="";
        $time_end="";
        $names[$y]=$name;

        $query_final="SELECT * FROM slices WHERE name='$name'";
        $result_final = mysql_query($query_final, $linkID) or die("Data not found.");
        $row_final = mysql_fetch_assoc($result_final);
        $owner1=$row_final['owner'];
        if (strcmp($owner1,$owner)==0){
            $slice->name=$row_final['name'];
            $slice->owner=$row_final['owner'];
            $time_start=$timestamp_start_check;
            $time_end=$timestamp_end_check;
            $slice->time_start=$time_start;
            $slice->time_end=$time_end;
            $slice_id=$row_final['slice_id'];
            $query1="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id}";
            $result1 = mysql_query($query1, $linkID) or die("Data not found.");

            include 'manifest.php';

            $sxe = new SimpleXMLElement($xmlstr);
            for($i = 0 ; $i < mysql_num_rows($result1) ; $i++){
                $row1 = mysql_fetch_assoc($result1);
                $node_id=$row1['node_id'];

                $query2="SELECT * FROM nodes WHERE id={$node_id}";
                $result2= mysql_query($query2, $linkID) or die("Data not found.");
                $row2 = mysql_fetch_assoc($result2);
                $node=$sxe->addChild('node');

                $id=$row2['id'];
                $node->addAttribute('client_id',"node".$id);
                $node->addAttribute('component_id',"urn:publicid:IDN+sorch.netnode.ntua.gr+node+node".$id);

```



```

    $node-
>addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
    $node->addAttribute('sliver_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+sliver+node".$id);
    $node->addAttribute('exclusive','true');

$hardware=$node->addChild('hardware_type');

$hardware->addAttribute('name','wireless access point');

$sliver=$node->addChild('sliver_type');

$name1="node".$id;
$sliver->addAttribute('name','access point');

    $query3= "SELECT * FROM node_if_association WHERE slice_id={$slice_id} AND
node_id={$node_id}";
    $result3 = mysql_query($query3, $linkID) or die("Data not found.");
    for($y = 0 ; $y < mysql_num_rows($result3) ; $y++){
        $interface=$node->addChild('interface');

        $row3 = mysql_fetch_assoc($result3);
        $if1=$row3['if_id'];
        $if=$name1.":".$if1;
        $interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
        $interface->addAttribute('client_id',$if);
        $wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        $array=array_keys($row3);

        for ($z=3;$z<count($array);$z++){
            $child=$wireless-
>addChild($array[$z],$row3[$array[$z]],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        }
    }
}
$string1=$sxe->asXML();

$slice->rspec=$string1;
$slicesarray[$x]=$slice;

$xdoc = new DomDocument();
$xmlschema = '/var/www/ws/RSpec/manifest.xsd';
//Load the xml document in the DOMDocument object
$xdoc->loadXML($string1);

if ($xdoc->schemaValidate($xmlschema)) {
    $validatearray[$x]=1;
}
else
{
    $validatearray[$x]=2;
}
}
}
}
$testbed->slicesarray=$slicesarray;
$testbed->validatearray=$validatearray;
return $testbed;
}

//list activated slices
function getListActivatedSlices(){

$host = "localhost";
$user = "omf";
$pass = "omf";
$database = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($database, $linkID) or die("Could not find database.");

```

```

$query="SELECT * FROM slices";
$result = mysql_query($query, $linkID) or die("Data not found.");

$testbed= new testbedslices;
$slicesarray=array();
$validatearray=array();
for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
    $row = mysql_fetch_assoc($result);
    $slice= new slice;
    $activate=$row['activate'];
    if ($activate==1)
    {

        $slice->name=$row['name'];
        $slice->owner=$row['owner'];
        $slice->time_start=$row['timeactivation'];
        $slice->duration=$row['duration_activation'];

        $slice_id=$row['slice_id'];

        $query1="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id}";
        $result1 = mysql_query($query1, $linkID) or die("Data not found.");

        $nodesarray=array();
        include 'manifest.php';

        $sxe = new SimpleXMLElement($xmlstr);
        for($i = 0 ; $i < mysql_num_rows($result1) ; $i++){
            $row1 = mysql_fetch_assoc($result1);
            $node_id=$row1['node_id'];

            $query2="SELECT * FROM nodes WHERE id={$node_id}";
            $result2= mysql_query($query2, $linkID) or die("Data not found.");
            $row2 = mysql_fetch_assoc($result2);
            $node=$sxe->addChild('node');

            $id=$row2['id'];
            $node->addAttribute('client_id',"node".$id);
            $node->addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+node+node".$id);
            $node-
            >addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
            $node->addAttribute('sliver_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+sliver+node".$id);
            $node->addAttribute('exclusive','true');

            $hardware=$node->addChild('hardware_type');

            $hardware->addAttribute('name','wireless access point');

            $sliver=$node->addChild('sliver_type');

            $name1="node".$id;
            $sliver->addAttribute('name','access point');

            $query3= "SELECT * FROM node_if_association WHERE slice_id={$slice_id} AND
            node_id={$node_id}";
            $result3 = mysql_query($query3, $linkID) or die("Data not found.");

            for($y = 0 ; $y < mysql_num_rows($result3) ; $y++){
                $interface=$node->addChild('interface');

                $row3 = mysql_fetch_assoc($result3);
                $if1=$row3['if_id'];
                $if=$name1.".".$if1;
                $interface-
                >addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
                $interface->addAttribute('client_id',$if);
                $wireless=$interface-
                >addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
                $array=array_keys($row3);

                for ($z=3;$z<count($array);$z++){
                    $child=$wireless-
                    >addChild($array[$z],$row3[$array[$z]],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");

```

```

}
}
}
$string1=$sxe->asXML();

$slice->rspec=$string1;
$slicesarray[$x]=$slice;

$xml = new DomDocument();
$xmlschema = '/var/www/ws/RSpec/manifest.xsd';
//Load the xml document in the DOMDocument object
$xml->loadXML($string1);

if ($xml->schemaValidate($xmlschema)) {
$validatearray[$x]=1;
}
else
{
$validatearray[$x]=2;
}
}
}
}
$testbed->slicearray=$slicesarray;
$testbed->validatearray=$validatearray;
return $testbed;
}

//list activated slices of a user
function getListMyActivatedSlices($owner){

$host = "localhost";
$user = "omf";
$pass = "omf";
$databse = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($databse, $linkID) or die("Could not find database.");

$query="SELECT * FROM slices WHERE owner='$owner'";
$result = mysql_query($query, $linkID) or die("Data not found.");

$testbed= new testbedslices;
$slicesarray=array();
$validatearray=array();
for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
    $row = mysql_fetch_assoc($result);
    $slice= new slice;
    $activate=$row['activate'];
    if ($activate==1)
    {
    $slice->name=$row['name'];
    $slice->owner=$row['owner'];
    $slice->time_start=$row['timeactivation'];
    $slice->duration=$row['duration_activation'];
    $slice_id=$row['slice_id'];

    $query1="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id}";
    $result1 = mysql_query($query1, $linkID) or die("Data not found.");

    $nodesarray=array();
    include 'manifest.php';
    $sxe = new SimpleXMLElement($xmlstr);
    for($i = 0 ; $i < mysql_num_rows($result1) ; $i++){
    $row1 = mysql_fetch_assoc($result1);
    $node_id=$row1['node_id'];

    $query2="SELECT * FROM nodes WHERE id={$node_id}";
    $result2= mysql_query($query2, $linkID) or die("Data not found.");
    $row2 = mysql_fetch_assoc($result2);
    $node=$sxe->addChild('node');

    $id=$row2['id'];
    $node->addAttribute('client_id',"node".$id);
    $node->addAttribute('component_id',"urn:publicid:IDN+sorch.netnode.ntua.gr+node+node".$id);

```

```

    $node-
>addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
    $node->addAttribute('sliver_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+sliver+node".$id);
    $node->addAttribute('exclusive','true');

$hardware=$node->addChild('hardware_type');

$hardware->addAttribute('name','wireless access point');

$sliver=$node->addChild('sliver_type');

$name1="node".$id;
$sliver->addAttribute('name','access point');

    $query3= "SELECT * FROM node_if_association WHERE slice_id={$slice_id} AND
node_id={$node_id}";
    $result3 = mysql_query($query3, $linkID) or die("Data not found.");

    for($y = 0 ; $y < mysql_num_rows($result3) ; $y++){
        $interface=$node->addChild('interface');

        $row3 = mysql_fetch_assoc($result3);
        $if1=$row3['if_id'];
        $if=$name1.".$if1";
        $interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
        $interface->addAttribute('client_id',$if);
        $wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        $array=array_keys($row3);

        for ($z=3;$z<count($array);$z++){
            $child=$wireless-
>addChild($array[$z],$row3[$array[$z]],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
        }
    }
}
}
}
$string1=$sxe->asXML();

$slice->rspec=$string1;
$slicesarray[$x]=$slice;

$xdoc = new DomDocument();
$xmlschema = '/var/www/ws/RSpec/manifest.xsd';
//Load the xml document in the DOMDocument object
$xdoc->loadXML($string1);

if ($xdoc->schemaValidate($xmlschema)) {
    $validatearray[$x]=1;
}
else
{
    $validatearray[$x]=2;
}
}
}
$testbed->slicearray=$slicesarray;
$testbed->validatearray=$validatearray;
return $testbed;
}

//list no-activated slices
function getListNoActivatedSlices(){

$host = "localhost";
$user = "omf";
$pass = "omf";
$databse = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($databse, $linkID) or die("Could not find database.");

```

```

$query="SELECT * FROM slices";
$result = mysql_query($query, $linkID) or die("Data not found.");

$testbed= new testbedslices;
$slicesarray=array();
$validatearray=array();
for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
    $row = mysql_fetch_assoc($result);
    $slice= new slice;
    $activate=$row['activate'];
    if ($activate==0)
    {

        $slice->name=$row['name'];
        $slice->owner=$row['owner'];
        $slice_id=$row['slice_id'];

        $query1="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id}";
        $result1 = mysql_query($query1, $linkID) or die("Data not found.");

        $nodesarray=array();
        include 'manifest.php';

        $sxe = new SimpleXMLElement($xmlstr);
        for($i = 0 ; $i < mysql_num_rows($result1) ; $i++){
            $row1 = mysql_fetch_assoc($result1);
            $node_id=$row1['node_id'];

            $query2="SELECT * FROM nodes WHERE id={$node_id}";
            $result2= mysql_query($query2, $linkID) or die("Data not found.");
            $row2 = mysql_fetch_assoc($result2);
            $node=$sxe->addChild('node');

            $id=$row2['id'];
            $node->addAttribute('client_id',"node".$id);
            $node->addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+node+node".$id);
            $node-
>addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
            $node->addAttribute('sliver_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+sliver+node".$id);
            $node->addAttribute('exclusive','true');

            $hardware=$node->addChild('hardware_type');

            $hardware->addAttribute('name','wireless access point');

            $sliver=$node->addChild('sliver_type');

            $name1="node".$id;
            $sliver->addAttribute('name','access point');;

            $query3= "SELECT * FROM node_if_association WHERE slice_id={$slice_id} AND
node_id={$node_id}";
            $result3 = mysql_query($query3, $linkID) or die("Data not found.");
            for($y = 0 ; $y < mysql_num_rows($result3) ; $y++){
                $interface=$node->addChild('interface');

                $row3 = mysql_fetch_assoc($result3);
                $if1=$row3['if_id'];
                $if=$name1.".".$if1;
                $interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface+".$if);
                $interface->addAttribute('client_id',$if);
                $wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
                $array=array_keys($row3);

                for ($z=3;$z<count($array);$z++){
                    $child=$wireless-
>addChild($array[$z],$row3[$array[$z]],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
                }
            }
        }
    }
}

```

```

$string1=$sxe->asXML();

$slice->rspec=$string1;
$slicesarray[$x]=$slice;

$xml = new DomDocument();
$xmlschema = '/var/www/ws/RSpec/manifest.xsd';
//Load the xml document in the DOMDocument object
$xml->loadXML($string1);

if ($xml->schemaValidate($xmlschema)) {
    $validatearray[$x]=1;
}
else
{
    $validatearray[$x]=2;
}
}
}
}
}
}
}
}
}
}
}

//list no-activated slices of user
function getListMyNoActivatedSlices($owner){

$host = "localhost";
$user = "omf";
$pass = "omf";
$databse = "inventory2";

$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($databse, $linkID) or die("Could not find database.");

$query="SELECT * FROM slices WHERE owner='$owner'";
$result = mysql_query($query, $linkID) or die("Data not found.");

$testbed= new testbedslices;
$slicesarray=array();
$validatearray=array();
for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
    $row = mysql_fetch_assoc($result);
    $slice= new slice;
    $activate=$row['activate'];
    if ($activate==0)
    {

        $slice->name=$row['name'];
        $slice->owner=$row['owner'];
        $slice_id=$row['slice_id'];
        $query1="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id}";
        $result1 = mysql_query($query1, $linkID) or die("Data not found.");

        $nodesarray=array();
        include 'manifest.php';

        $sxe = new SimpleXMLElement($xmlstr);
        for($i = 0 ; $i < mysql_num_rows($result1) ; $i++){
            $row1 = mysql_fetch_assoc($result1);
            $node_id=$row1['node_id'];

            $query2="SELECT * FROM nodes WHERE id={$node_id}";
            $result2= mysql_query($query2, $linkID) or die("Data not found.");
            $row2 = mysql_fetch_assoc($result2);
            $node=$sxe->addChild('node');

            $id=$row2['id'];
            $node->addAttribute('client_id',"node".$id);
            $node->addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+node+node".$id);
            $node-
            >addAttribute('component_manager_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+authority+cm");
            $node->addAttribute('sliver_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+sliver+node".$id);

```

```

        $node->addAttribute('exclusive','true');

$hardware=$node->addChild('hardware_type');

$hardware->addAttribute('name','wireless access point');

$sliver=$node->addChild('sliver_type');

$name1="node".$id;
$sliver->addAttribute('name','access point');

        $query3= "SELECT * FROM node_if_association WHERE slice_id={ $slice_id } AND
node_id={ $node_id }";
        $result3 = mysql_query($query3, $linkID) or die("Data not found.");
        for($y = 0 ; $y < mysql_num_rows($result3) ; $y++){
            $interface=$node->addChild('interface');

            $row3 = mysql_fetch_assoc($result3);
            $if1=$row3['if_id'];
            $if=$name1.".$if1;
            $interface-
>addAttribute('component_id',"urn:publicid:IDN+sorch.netmode.ntua.gr+interface".$if);
            $interface->addAttribute('client_id',$if);
            $wireless=$interface-
>addChild('Wireless_features',NULL,"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
            $array=array_keys($row3);

            for ($z=3;$z<count($array);$z++){
                $child=$wireless-
>addChild($array[$z],$row3[$array[$z]],"http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
            }
        }
    }
}
$string1=$sxe->asXML();
$slice->rspec=$string1;
$slicesarray[$x]=$slice;
$xdoc = new DomDocument();
$xmlschema = '/var/www/ws/RSpec/manifest.xsd';
//Load the xml document in the DOMDocument object
$xdoc->loadXML($string1);

if ($xdoc->schemaValidate($xmlschema)) {
    $validatearray[$x]=1;
}
else
{
    $validatearray[$x]=2;
}
}
}
}
$testbed->slicearray=$slicesarray;
$testbed->validatearray=$validatearray;
return $testbed;
}

ini_set('soap.wsdl_cache_enabled', '0');
$server = new SoapServer("listslices12.wsdl");
$server->addFunction("getListSlices");
$server->addFunction("getListMySlices");
$server->addFunction("getListAvailableResources");
$server->addFunction("getListActivatedSlices");
$server->addFunction("getListMyActivatedSlices");
$server->addFunction("getListNoActivatedSlices");
$server->addFunction("getListMyNoActivatedSlices");
$server->addFunction("getListNoReservedSlices");
$server->addFunction("getListMyNoReservedSlices");
$server->addFunction("getListReservedSlices");
$server->addFunction("getListMyReservedSlices");
$server->addFunction("getAvailableResources");
$server->addFunction("getListSliceResources");
$server->handle();

?>

```

listslices.wsdl

```
<?xml version='1.0' encoding='UTF-8' ?>

<definitions name='Slices'
  targetNamespace='http://sorch.netmode.ntua.gr/ws'
  xmlns:tns='http://sorch.netmode.ntua.gr/ws'
  xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
  xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'
  xmlns:xsd1="http://sorch.netmode.ntua.gr/ws"
  xmlns='http://schemas.xmlsoap.org/wsdl/'>

  <types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://sorch.netmode.ntua.gr/ws"
      targetNamespace="http://sorch.netmode.ntua.gr/ws" xmlns:xsd1="http://sorch.netmode.ntua.gr/ws">

      <complexType name="slice">
        <sequence>
          <element name="name" type="xsd:string"/>
          <element name="owner" type="xsd:string"/>
          <element name="time_start" type="xsd:string"/>
          <element name="time_end" type="xsd:string"/>
          <element name="duration" type="xsd:string"/>
          <element name="rspec" type="xsd:string"/>
        </sequence>
      </complexType>

      <complexType name="slicearray">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd1:slice[]" />
          </restriction>
        </complexContent>
      </complexType>

      <complexType name="validatearray">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:int[]" />
          </restriction>
        </complexContent>
      </complexType>

      <complexType name="testbedslices">
        <sequence>
          <element name="slicearray" type="xsd1:slicearray"/>
          <element name="validatearray" type="xsd1:validatearray"/>
        </sequence>
      </complexType>
    </schema>
  </types>

  <message name='getAvailableResourcesRequest'>
    <part name='Result' type='xsd:string' />
  </message>

  <message name='getAvailableResourcesResponse'>
    <part name='Result' type='xsd:string' />
  </message>

  <message name='getListSlicesRequest'>
  </message>

  <message name='getListSlicesResponse'>
    <part name='Result' type='xsd1:testbedslices' />
  </message>

  <message name='getListMySlicesRequest'>
    <part name='Result' type='xsd:string' />
  </message>

  <message name='getListMySlicesResponse'>
    <part name='Result' type='xsd1:testbedslices' />
  </message>

```



```

<message name='getListActivatedSlicesRequest'>
</message>

<message name='getListActivatedSlicesResponse'>
  <part name='Result' type='xsd:testbedslices' />
</message>

<message name='getListMyActivatedSlicesRequest'>
<part name='Result' type='xsd:string' />
</message>

<message name='getListMyActivatedSlicesResponse'>
  <part name='Result' type='xsd:testbedslices' />
</message>

<message name='getListNoActivatedSlicesRequest'>
</message>

<message name='getListNoActivatedSlicesResponse'>
  <part name='Result' type='xsd:testbedslices' />
</message>

<message name='getListMyNoActivatedSlicesRequest'>
<part name='Result' type='xsd:string' />
</message>

<message name='getListMyNoActivatedSlicesResponse'>
  <part name='Result' type='xsd:testbedslices' />
</message>

<message name='getListReservedSlicesRequest'>
</message>

<message name='getListReservedSlicesResponse'>
  <part name='Result' type='xsd:testbedslices' />
</message>

<message name='getListMyReservedSlicesRequest'>
<part name='Result' type='xsd:string' />
</message>

<message name='getListMyReservedSlicesResponse'>
  <part name='Result' type='xsd:testbedslices' />
</message>

<message name='getListNoReservedSlicesRequest'>
</message>

<message name='getListNoReservedSlicesResponse'>
  <part name='Result' type='xsd:testbedslices' />
</message>

<message name='getListMyNoReservedSlicesRequest'>
<part name='Result' type='xsd:string' />
</message>

<message name='getListMyNoReservedSlicesResponse'>
  <part name='Result' type='xsd:testbedslices' />
</message>

<message name='getListAvailableResourcesRequest'>
<part name='Result' type='xsd:string' />
</message>

<message name='getListAvailableResourcesResponse'>
  <part name='Result' type='xsd:string' />
</message>

<message name='getListSliceResourcesRequest'>
<part name='Result' type='xsd:string' />
</message>

<message name='getListSliceResourcesResponse'>
  <part name='Result' type='xsd:string' />
</message>

```

```

<portType name='ListSlicesPortType'>
  <operation name='getListSlices'>
    <input message='tns:getListSlicesRequest' />
    <output message='tns:getListSlicesResponse' />
  </operation>

  <operation name='getListMySlices'>
    <input message='tns:getListMySlicesRequest' />
    <output message='tns:getListMySlicesResponse' />
  </operation>

  <operation name='getListActivatedSlices'>
    <input message='tns:getListActivatedSlicesRequest' />
    <output message='tns:getListActivatedSlicesResponse' />
  </operation>

  <operation name='getListMyActivatedSlices'>
    <input message='tns:getListMyActivatedSlicesRequest' />
    <output message='tns:getListMyActivatedSlicesResponse' />
  </operation>

  <operation name='getListNoActivatedSlices'>
    <input message='tns:getListNoActivatedSlicesRequest' />
    <output message='tns:getListNoActivatedSlicesResponse' />
  </operation>

  <operation name='getListMyNoActivatedSlices'>
    <input message='tns:getListMyNoActivatedSlicesRequest' />
    <output message='tns:getListMyNoActivatedSlicesResponse' />
  </operation>

  <operation name='getListReservedSlices'>
    <input message='tns:getListReservedSlicesRequest' />
    <output message='tns:getListReservedSlicesResponse' />
  </operation>

  <operation name='getListMyReservedSlices'>
    <input message='tns:getListMyReservedSlicesRequest' />
    <output message='tns:getListMyReservedSlicesResponse' />
  </operation>

  <operation name='getListNoReservedSlices'>
    <input message='tns:getListNoReservedSlicesRequest' />
    <output message='tns:getListNoReservedSlicesResponse' />
  </operation>

  <operation name='getListMyNoReservedSlices'>
    <input message='tns:getListMyNoReservedSlicesRequest' />
    <output message='tns:getListMyNoReservedSlicesResponse' />
  </operation>

  <operation name='getListAvailableResources'>
    <input message='tns:getListAvailableResourcesRequest' />
    <output message='tns:getListAvailableResourcesResponse' />
  </operation>

  <operation name='getAvailableResources'>
    <input message='tns:getAvailableResourcesRequest' />
    <output message='tns:getAvailableResourcesResponse' />
  </operation>

  <operation name='getListSliceResources'>
    <input message='tns:getListSliceResourcesRequest' />
    <output message='tns:getListSliceResourcesResponse' />
  </operation>
</portType>

<binding name='ListSlicesBinding' type=
'tns:ListSlicesPortType'>
  <soap:binding style='rpc'
  transport='http://schemas.xmlsoap.org/soap/http'
  />

  <operation name='getListSlices'>

```

```

<soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getListSlices' />
<input>
  <soap:body use='encoded' namespace=
    'http://sorch.netmode.ntua.gr/ws'
    encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
</input>
<output>
  <soap:body use='encoded' namespace=
'http://sorch.netmode.ntua.gr/ws'
    encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
</output>
</operation>

<operation name='getListMySlices'>
  <soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getListMySlices' />
  <input>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </input>
  <output>
    <soap:body use='encoded' namespace=
'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </output>
</operation>

<operation name='getListActivatedSlices'>
  <soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getListActivatedSlices' />
  <input>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </input>
  <output>
    <soap:body use='encoded' namespace=
'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </output>
</operation>

<operation name='getListMyActivatedSlices'>
  <soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getListMyActivatedSlices' />
  <input>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </input>
  <output>
    <soap:body use='encoded' namespace=
'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </output>
</operation>

<operation name='getListNoActivatedSlices'>
  <soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getListNoActivatedSlices' />
  <input>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </input>
  <output>
    <soap:body use='encoded' namespace=
'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </output>
</operation>

<operation name='getListMyNoActivatedSlices'>
  <soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getListMyNoActivatedSlices' />
  <input>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </input>
  <output>
    <soap:body use='encoded' namespace=

```

```

'http://sorch.netmode.ntua.gr/ws'
  encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
</output>
</operation>

<operation name='getListReservedSlices'>
  <soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getListReservedSlices'/>
  <input>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
    </input>
  <output>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
    </output>
  </operation>

<operation name='getListMyReservedSlices'>
  <soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getListMyReservedSlices'/>
  <input>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
    </input>
  <output>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
    </output>
  </operation>

<operation name='getListNoReservedSlices'>
  <soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getListNoReservedSlices'/>
  <input>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
    </input>
  <output>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
    </output>
  </operation>

<operation name='getListMyNoReservedSlices'>
  <soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getListMyNoReservedSlices'/>
  <input>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
    </input>
  <output>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
    </output>
  </operation>

<operation name='getListAvailableResources'>
  <soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getListAvailableResources'/>
  <input>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
    </input>
  <output>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
    </output>
  </operation>

```

```

<operation name='getAvailableResources'>
  <soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getAvailableResources' />
  <input>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </input>
  <output>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </output>
</operation>

<operation name='getListSliceResources'>
  <soap:operation soapAction='http://sorch.netmode.ntua.gr/ws/getListSliceResources' />
  <input>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </input>
  <output>
    <soap:body use='encoded' namespace=
      'http://sorch.netmode.ntua.gr/ws'
      encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
  </output>
</operation>

</binding>

<service name='ListSlicesService2'>
  <port name='ListSlicesPortType' binding=
    'ListSlicesBinding'>
    <soap:address location='http://sorch.netmode.ntua.gr/ws/listslicesservicenew2.php' />
  </port>
</service>
</definitions>

```

listmyslices.php

```

<?php
//list slices of a user
session_start();
$owner=$_SESSION['username'];

Echo "<html>";
Echo "<title>List my slices</title>";

$client = new SoapClient("listslices12.wsdl");
$response1 = $client->getListMySlices($owner);
$response2=$response1->validatearray;

if(in_array(2,$response2))
{
echo "not valid rspec";
}
else
{
Echo "<h1>List My Slices</h1>";

$response=$response1->slicearray;
Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner." contains the following
nodes of testbed: </li> ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];
$array=split(":",$component_id);
$string1=$array[2];

```

```

$array1=split("\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":", $check1);
$if_id=$array[1];
Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName, 'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}

Echo "</ul>";
}
echo "<a href='\http://sorch.netmode.ntua.gr/ws/services.php\'>Return to the services</a><br/>";
Echo "</html>";
?>

```

listMyNoReservedSlice.php

```

<?php
//list no-reserved slices of a user
session_start();
$owner=$_SESSION['username'];
Echo "<html>";
Echo "<title>List no-reserved slices of testbed</title>";
$client = new SoapClient("listslices12.wsdl");
$client1 = new SoapClient("listslices12.wsdl");
$client2 = new SoapClient("listslices12.wsdl");
$response3 = $client->getListMyActivatedSlices($owner);
$response5 = $client1->getListMyReservedSlices($owner);
$response1 = $client2->getListMyNoReservedSlices($owner);
$response2=$response1->validatearray;
$response6=$response3->validatearray;
$response4=$response5->validatearray;

if((in_array(2,$response2)) || (in_array(2,$response4)) || (in_array(2,$response6)))
{
echo "not valid rspec";
}
else
{
$response=$response5->slicearray;
$reserved=array();
for($x = 0 ; $x < count($response) ; $x++){

```

```

$reserved[]=$response[$x]->name;
}

Echo "<h1>List of no-reserved Slices</h1>";
$response=$response1->slicearray;
$check_array=$response3->slicearray;
if ((count($response)==0) && (count($check_array)==0))
{
echo "There is not any no-reserved slice <br/>";
}
else{
Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner." contains the following
nodes of testbed: </li> ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];
$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$sif_id=$array[1];
Echo "<li>interface ".$sif_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}

Echo "</ul>";
}

$response=$response3->slicearray;
$check_array=$response1->slicearray;
if ((count($response)==0) && (count($check_array)==0))
{
}
else{

```

```

Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
if(!in_array($response[$x]->name,$reserved)){
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner." ,which has been
activated from ".$response[$x]->time_start." and for ".$response[$x]->duration." hours contains
the following nodes of testbed: </li> ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];
$array=split(":",$component_id);
$string1=$array[2];
$array1=split("\",$string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;

Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check1=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}
}
Echo "</ul>";
}
}
echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
Echo "</html>";
?>

```

listMyReservedSlices.php

```

<?php
session_start();
$owner=$_SESSION['username'];
Echo "<html>";
Echo "<title>List my reserved slices</title>";
$client = new SoapClient("listslices12.wsdl");
$response1 = $client->getListMyReservedSlices($owner);
$response2=$response1->validatearray;

```



```

if(in_array(2,$response2))
{
echo "not valid rspec";
}
else
{

Echo "<h1>List of Reserved Slices</h1>";
$response=$response1->slicearray;

if (count($response)==0)
{
echo "There is not any reserved slice <br/>";
}
else{
Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
$time_start=$response[$x]->time_start;
$array_start=split(",",$time_start);

$time_end=$response[$x]->time_end;
$array_end=split(",",$time_end);

Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner;
Echo " has been reserved ";
for($y = 0 ; $y < count($array_start) ; $y++){

if ($y==0)
{
Echo " from ".$array_start[$y]." to ".$array_end[$y];
}
else
{
Echo " and from ".$array_start[$y]." to ".$array_end[$y];
}
}

Echo ".The slice contains the following nodes of testbed: </li>";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){
$component_id=$entry2['component_id'];
$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];
Echo "<li>interface ".$if_id." with ";
$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();
$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)

```

```

{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}
Echo "</ul>";
}
}
echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
Echo "</html>";
?>

```

listMyNoActivatedSlices.php

```

<?php
//list no-activated slices of a user and divide them in two states
session_start();
$owner=$_SESSION['username'];
Echo "<html>";
Echo "<title>List my no-activated slices</title>";
$client = new SoapClient("listslices12.wsdl");
$client1 = new SoapClient("listslices12.wsdl");
$client2 = new SoapClient("listslices12.wsdl");
$response5 = $client->getListMyActivatedSlices($owner);
$response4=$response5->validatearray;
$response3 = $client1->getListMyReservedSlices($owner);
$response1 = $client2->getListMyNoReservedSlices($owner);
$response2=$response1->validatearray;
$response6=$response3->validatearray;
if((in_array(2,$response2)) || (in_array(2,$response4)) || (in_array(2,$response6)))
{
echo "not valid rspec";
}
else
{
$response=$response5->slicearray;
$reserved=array();
for($x = 0 ; $x < count($response) ; $x++){
$reserved[]=$response[$x]->name;
}
Echo "<h1>List of no-activated Slices</h1>";
$response=$response1->slicearray;
Echo "<p><b>List of no-reserved Slices which are not activate now</b></p>";
if (count($response)==0)
{
echo "There is not any noreserved slice that is noactivated <br/>";
}
else{
Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner." contains the following
nodes of testbed: </li> ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);
Echo "<ol>";
foreach ($simplexml->node as $entry2){
$component_id=$entry2['component_id'];
$array=split(":",$component_id);
$string1=$array[2];
$array1=split("\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{

```

```

Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];
Echo "<li>interface ".$if_id." with ";
$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();
$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}

Echo "</ul>";
}

$response=$response3->slicearray;
Echo "<p><b>List of reserved Slices which are not activate now</b></p>";
if (count($response)==0)
{
echo "There is not any reserved slice that is noactivated now <br/>";
}
else{
Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
if(!in_array($response[$x]->name,$reserved)){
$time_start=$response[$x]->time_start;
$array_start=split(",",$time_start);
$time_end=$response[$x]->time_end;
$array_end=split(",",$time_end);
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner;
Echo " has been reserved ";
for($y = 0 ; $y < count($array_start) ; $y++){
if ($y==0)
{
Echo " from ".$array_start[$y]." to ".$array_end[$y];
}
else
{
Echo " and from ".$array_start[$y]." to ".$array_end[$y];
}
}
Echo ".The slice contains the following nodes of testbed: </li>";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);
Echo "<ol>";
foreach ($simplexml->node as $entry2){
$component_id=$entry2['component_id'];
$array=split(":",$component_id);
$string1=$array[2];
$array1=split("\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)

```

```

{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interface1['client_id'];
$array=split(":",$check1);
$if_id=$array[1];
Echo "<li>interface ".$if_id." with ";

$dc2 = $interface1->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}
}
Echo "</ul>";
}
}
echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
Echo "</html>";
?>

```

listMyActivatedSlices.php

```

<?php
//list activated slices of a user
session_start();
$owner=$_SESSION['username'];
Echo "<html>";
Echo "<title>List my activated slices</title>";
$client = new SoapClient("listslices12.wsdl");
$response1 = $client->getListMyActivatedSlices($owner);
$response2=$response1->validatearray;

if(in_array(2,$response2)
{
echo "not valid rspec";
}
else
{
Echo "<h1>List of Activated Slices</h1>";
$response=$response1->slicearray;
if (count($response)==0)
{
echo "There is not any activated slice <br/>";
}
else{
Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner." ,which has been
activated from ".$response[$x]->time_start." and for ".$response[$x]->duration." hours contains
the following nodes of testbed: </li> ";

$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";

```

```

foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];
$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":", $check1);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}
Echo "</ul>";
}
}
echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
Echo "</html>";
?>

```

listslices.php

```

<?php
//list all slices of testbed
Echo "<html>";
Echo "<title>List all slices of testbed</title>";

$client = new SoapClient("listslices12.wsdl");
$response1 = $client->getListSlices();
$response2=$response1->validatearray;

if(in_array(2,$response2))
{
echo "not valid rspec";
}
else
{
Echo "<h1>List of Slices</h1>";

$response=$response1->slicearray;
Echo "<ul>";

```

```

for($x = 0 ; $x < count($response) ; $x++){
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner." contains the following
nodes of testbed: </li> ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){
$component_id=$entry2['component_id'];
$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":", $check1);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}

Echo "</ul>";
}
echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
Echo "</html>";
?>

```

listNoReservedSlices.php

```

<?php
//list no-reserved slices of testbed
Echo "<html>";
Echo "<title>List no-reserved slices of testbed</title>";

$client = new SoapClient("listslices12.wsdl");
$client1 = new SoapClient("listslices12.wsdl");
$client2 = new SoapClient("listslices12.wsdl");
$response3 = $client->getListActivatedSlices();
$response5 = $client1->getListReservedSlices();
$response1 = $client2->getListNoReservedSlices();
$response2=$response1->validatearray;
$response6=$response3->validatearray;
$response4=$response5->validatearray;

```

```

if((in_array(2,$response2) || (in_array(2,$response4) || (in_array(2,$response6)))
{
echo "not valid rspec";
}
else
{
$response=$response5->slicearray;
$reserved=array();
for($x = 0 ; $x < count($response) ; $x++){
$reserved[]=$response[$x]->name;
}

Echo "<h1>List of no-reserved Slices</h1>";
$response=$response1->slicearray;
$check_array=$response3->slicearray;
if ((count($response)==0) && (count($check_array)==0))
{
echo "There is not any noreserved slice <br/>";
}
else{
Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner." contains the following
nodes of testbed: </li> ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);
Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;

Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
}
Echo "</ul>";
}
Echo "<br/>";
}

```

```

    Echo "</ol>";
}

Echo "</ul>";
}

$response=$response3->slicearray;
$check_array=$response1->slicearray;
if ((count($response)==0) && (count($check_array)==0))
{
}
else{
Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
if(!in_array($response[$x]->name,$reserved)){
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner." ,which has been
activated from ".$response[$x]->time_start." and for ".$response[$x]->duration." hours contains
the following nodes of testbed: </li> ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){
$component_id=$entry2['component_id'];
$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}
}
Echo "</ul>";
}
}
echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
Echo "</html>";
?>

```


listReservedSlices.php

```
<?php
//list reserved slices of testbed
Echo "<html>";
Echo "<title>List reserved slices of testbed</title>";

$client = new SoapClient("listslices12.wsdl");
$response1 = $client->getListReservedSlices();
$response2=$response1->validatearray;
if(in_array(2,$response2)
{
echo "not valid rspec";
}
else
{

Echo "<h1>List of Reserved Slices</h1>";
$response=$response1->slicearray;

if (count($response)==0)
{
echo "There is not any reserved slice <br/>";
}
else{
Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
$time_start=$response[$x]->time_start;
$array_start=split(",",$time_start);

$time_end=$response[$x]->time_end;
$array_end=split(",",$time_end);
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner;
Echo " has been reserved ";
for($y = 0 ; $y < count($array_start) ; $y++){

if ($y==0)
{
Echo " from ".$array_start[$y]." to ".$array_end[$y];
}
else
{
Echo " and from ".$array_start[$y]." to ".$array_end[$y];
}
}

Echo ".The slice contains the following nodes of testbed: </li>";

$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\\+",$string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check1=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
```

```

$if_id=$array[1];
Echo "<li>interface ".$if_id." with ";

$dc2 = $interface1->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}

Echo "</ul>";
}

}
echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
Echo "</html>";
?>

```

listNoActivatedSlices.php

```

<?php
//list all no activated slices divided in two states
Echo "<html>";
Echo "<title>List no-activated slices of testbed</title>";

$client = new SoapClient("listslices12.wsdl");
$client1 = new SoapClient("listslices12.wsdl");
$client2 = new SoapClient("listslices12.wsdl");
$response5 = $client->getListActivatedSlices();
$response4=$response5->validatearray;
$response3 = $client1->getListReservedSlices();
$response1 = $client2->getListNoReservedSlices();
$response2=$response1->validatearray;
$response6=$response3->validatearray;

if((in_array(2,$response2)) || (in_array(2,$response4)) || (in_array(2,$response6)))
{
echo "not valid rspec";
}
else
{
$response=$response5->slicearray;
$reserved=array();
for($x = 0 ; $x < count($response) ; $x++){
$reserved[]=$response[$x]->name;
}
Echo "<h1>List of no-activated Slices</h1>";
$response=$response1->slicearray;
Echo "<p><b>List of no-reserved Slices which are not activate now</b></p>";
if (count($response)==0)
{
echo "There is not any noreserved slice that is noactivated <br/>";
}
else{
Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner." contains the following nodes of testbed: </li> ";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

```

```

Echo "<ol>";
foreach ($simplexml->node as $entry2){
$component_id=$entry2['component_id'];
$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\+",$string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];
Echo "<li>interface ".$if_id." with ";
$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}

Echo "</ul>";
}

$response=$response3->slicearray;
Echo "<p><b>List of reserved Slices which are not activate now</b></p>";
if (count($response)==0)
{
echo "There is not any reserved slice that is noactivated now <br/>";
}
else{
Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
if(!in_array($response[$x]->name,$reserved)){
$time_start=$response[$x]->time_start;
$array_start=split(":",$time_start);
$time_end=$response[$x]->time_end;
$array_end=split(":",$time_end);
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner;
Echo " has been reserved ";
for($y = 0 ; $y < count($array_start) ; $y++){

if ($y==0)
{
Echo " from ".$array_start[$y]." to ".$array_end[$y];
}
else
{
Echo " and from ".$array_start[$y]." to ".$array_end[$y];
}
}
}
}
}
}

```

```

}
}

Echo ".The slice contains the following nodes of testbed: </li>";
$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];
$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];
Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}
Echo "</ul>";
}
}
echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
Echo "</html>";
?>

```

listactivatedslicesnew.php

```

<?php
Echo "<html>";
Echo "<title>List activated slices of testbed</title>";

$client = new SoapClient("listslices12.wsdl");
$response1 = $client->getListActivatedSlices();
$response2=$response1->validatearray;

if(in_array(2,$response2))
{

```

```

echo "not valid rspec";
}
else
{

Echo "<h1>List of Activated Slices</h1>";
$response=$response1->slicearray;

if (count($response)==0)
{
echo "There is not any activated slice <br/>";
}
else{
Echo "<ul>";
for($x = 0 ; $x < count($response) ; $x++){
Echo "<li>The ".$response[$x]->name." with owner ".$response[$x]->owner." ,which has been
activated from ".$response[$x]->time_start." and for ".$response[$x]->duration." hours contains
the following nodes of testbed: </li> ";

$rspec=$response[$x]->rspec;
$simplexml = new SimpleXMLElement($rspec);

Echo "<ol>";
foreach ($simplexml->node as $entry2){

$component_id=$entry2['component_id'];

$array=split(":",$component_id);
$string1=$array[2];

$array1=split("\\+", $string1);
$hostname=$array1[3];
Echo "<li>The ".$hostname;
Echo " with interfaces: </li>";
Echo "<ul>";
foreach ($entry2->interface as $interfacel)
{
if (!$interfacel)
{
Echo " with no interfaces </li>";
$check1=1;
}
else
{
$check1=$interfacel['client_id'];
$array=split(":",$check1);
$if_id=$array[1];

Echo "<li>interface ".$if_id." with ";

$dc2 = $interfacel->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc2->Wireless_features as $entry3){
$dc1 = $entry3->children("http://sorch.netmode.ntua.gr/ws/RSpec/ext/wireless_if");
foreach ($dc1 as $features){
$nodeName=$features->getName();

$value=$entry3->$nodeName;
Echo " , ".$nodeName." = ".$value;
if (strcmp($nodeName,'Rate')==0)
{
Echo " Mbps ";
}
}
}
Echo "</li>";
}
Echo "</ul>";
}
Echo "<br/>";
Echo "</ol>";
}
Echo "</ul>";
}
}
}

```

```

echo "<a href=\"http://sorch.netmode.ntua.gr/ws/services.php\">Return to the services</a><br/>";
Echo "</html>";
?>

```

Το παρακάτω php script είναι αυτό που τρέχει κάθε μία ώρα στο μηχανήμα sorch.netmode.ntua.gr για να εντοπίσει ποιά slices είναι ενεργοποιημένα και ποιοί πόροι και με τι χαρακτηριστικά έχουν εκχωρηθεί σε αυτά. Με κατάλληλο bash script φορτώνει τα SSH public keys του ερευνητή στους κόμβους που έχουν εκχωρηθεί και με χρήση άλλων bash scripts ορίζει τα χαρακτηριστικά μετάδοσης των ασύρματων διεπαφών που έχουν εκχωρηθεί στο slice. Παραθέτουμε παρακάτω τα αντίστοιχα scripts:

activate.php

```

#!/usr/bin/php5
<?php
$host = "localhost";
$user = "omf";
$pass = "omf";
$database = "inventory2";
$linkID = mysql_connect($host, $user, $pass) or die("Could not connect to host.");
mysql_select_db($database, $linkID) or die("Could not find database.");
$query="SELECT * FROM slices";
$result = mysql_query($query, $linkID) or die("Data not found.");
$node_check=array();
for($x = 0 ; $x < mysql_num_rows($result) ; $x++){
    $row = mysql_fetch_assoc($result);
    $activate=$row['activate'];
    if ($activate==1)
    {
        $name=$row['name'];
        $slice_id=$row['slice_id'];
        $owner=$row['owner'];

        $query1="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id}";
        $result1 = mysql_query($query1, $linkID) or die("Data not found.");
        for($i = 0 ; $i < mysql_num_rows($result1) ; $i++){
            $row1 = mysql_fetch_assoc($result1);
            $node_id=$row1['node_id'];
            $node_check[]=$node_id;
            $hostname="node".$node_id;
            $foo=`/var/www/ws/key_add.sh '$hostname' '$owner'`;
            $query3="SELECT * FROM node_if_association WHERE slice_id={$slice_id} AND
node_id={$node_id}";
            $result3 = mysql_query($query3, $linkID) or die("Data not found.");
            for($z = 0 ; $z < mysql_num_rows($result3) ; $z++){
                $row3 = mysql_fetch_assoc($result3);
                $if=$row3['if_id'];
                $if1=preg_replace("/[^0-9s]/", "", $if);
                $rate=$row3['Rate'];
                $channel=$row3['RadioChannel'];
                $mode=$row3['OperatingMode'];
                if (strcmp($mode,"AP")==0)
                {
                    $foo=`/var/www/ws/set_AP.sh '$node_id' '$if1' '$name' '$channel' '$rate'`;
                }
                if (strcmp($mode,"STA")==0)
                {
                    $foo=`/var/www/ws/set_STA.sh '$node_id' '$if1' '$name' '$channel' '$rate'`;
                }
            }
        }
    }
    else
    {
        $name=$row['name'];
        $slice_id=$row['slice_id'];

        $query1="SELECT node_id FROM slice_node_association WHERE slice_id={$slice_id}";
        $result1 = mysql_query($query1, $linkID) or die("Data not found.");
    }
}

```

```

for($i = 0 ; $i < mysql_num_rows($result1) ; $i++){
$row1 = mysql_fetch_assoc($result1);
$node_id=$row1['node_id'];
if (!in_array($node_id,$node_check))
{
$hostname="node".$node_id;
if (($node_id==6) || ($node_id==10))
{
$foo=`/var/www/ws/key_del.sh '$hostname'`;
}
}
}
}
?>

```

key_add.sh

```

#!/bin/sh

NODE=$1
USER=$2
/bin/cat /id_rsa.pub > /auth.temp
/bin/cat /var/www/ws/keys/$USER.pub >> /auth.temp
/usr/bin/scp /auth.temp root@$NODE:/root/.ssh/authorized_keys > /dev/null 2>&1
/bin/rm -f /auth.temp

```

key_del.sh

```

#!/bin/sh

NODE=$1
/usr/bin/scp /id_rsa.pub root@$NODE:/root/.ssh/authorized_keys > /dev/null 2>&1

```

set_AP.sh

```

#!/bin/sh

node_id=$1
inf=$2
slice_name=$3
channel=$4
rate=$5

ssh root@node$node_id '
ifconfig ath'$inf' down 2> /dev/null;
wlanconfig ath'$inf' destroy 2> /dev/null;

wlanconfig ath'$inf' create wlandev wifi'$inf' wlanmode ap;
iwpriv ath'$inf' bgscan 0;
iwconfig ath'$inf' channel '$channel';
iwconfig ath'$inf' essid "'$slice_name'";
ifconfig ath'$inf' up 192.168.3.'$node_id';
iwconfig ath'$inf' rate '${rate}'M;
'

```

set_STA.sh

```

#!/bin/sh

node_id=$1
inf=$2
slice_name=$3
channel=$4
rate=$5

ssh root@node$node_id '
ifconfig ath'$inf' down 2> /dev/null;
wlanconfig ath'$inf' destroy 2> /dev/null;

wlanconfig ath'$inf' create wlandev wifi'$inf' wlanmode sta;
iwpriv ath'$inf' bgscan 0;
ifconfig ath'$inf' up;
modprobe wlan_scan_sta;
iwconfig ath'$inf' essid "'$slice_name'";
ifconfig ath'$inf' up 192.168.3.'$node_id';
iwconfig ath'$inf' rate '${rate}'M;
'

```