

# **ŽILINSKÁ UNIVERZITA V ŽILINE**

**Fakulta riadenia a informatiky**

## **DIPLOMOVÁ PRÁCA**

**Študijný odbor: Informačné systémy**

**Zameranie: Informačno-komunikačné siete**

**Bc. Martin Šanko**

**Webová aplikácia na podporu výučby študentov,  
ako rozšírenie možností nástroja dynamips a dynagen.**

**Vedúci: Ing. Pavel Segeč PhD.**

**Reg. č.: 400/2009      Máj 2010**

ŽILINSKÁ UNIVERZITA V ŽILINE, FAKULTA RIADENIA A INFORMATIKY

ZADANIE TÉMY DIPLOMOVEJ PRÁCE

Študijný odbor: Informačné systémy  
Zameranie: Informačno-komunikačné siete

Meno a priezvisko

Martin Šanko

Osobné číslo

551490

Názov práce v slovenskom aj anglickom jazyku

Webová aplikácia na podporu výučby študentov, ako rozšírenie možnosti nástroja dynamips a dynagen.  
Web application for the student learning support as a capabilities extension of the dynamips and dynagen tools.

Zadanie úlohy, ciele, pokyny pre vypracovanie

**Cieľ diplomovej práce:**

Cieľom DP je vyvinúť on-line, cez internet dostupné riešenie, ktoré by umožňovalo študentom, nezávisle od OS ich PC, vzdialený prístup k serveru na podporu vyučovania. Na servery bude voľne šíriteľný a platformovo nezávislý simulátor počítačových sietí. Web aplikácia umožní prístup pre vzdialených používateľov, ktorí by si mohli takto odskúšať reálnu sieťovú topológiu.

**Obsah:**

Pri riešení sa zamerajte na:

- \* Analýza správania a možností nástroja Dynamips/Dynagen.
- \* špecifikácia riešenia z pohľadu používateľov a vlastností.
- \* Návrh riešenia, výber spôsobu implementácie a implementácia.
- \* Testovanie a dokumentácia.
- \* Nasadenie.

Témy z predmetov študijného zamerania

Meno a pracovisko vedúceho DP:

Ing. Pavel Segeč PhD., FRI - KIS, ŽU

Meno a pracovisko tútora DP:

vedúci DP  
(dátum a podpis)

tútor  
(dátum a podpis)

vedúci katedry  
(dátum a podpis)

garant  
(dátum a podpis)

Zadanie zaregistrované dňa 02. 11. 2009 pod číslom 400/2009 podpis \_\_\_\_\_

## **Abstrakt**

ŠANKO, Martin: Webová aplikácia na podporu výučby študentov, ako rozšírenie možností nástroja Dynamips a Dynagen [diplomová práca] - Žilinská univerzita v Žiline, Fakulta riadenia a informatiky; Katedra informačných sietí. – Vedúci: Ing. Pavel Segeč, PhD. – Stupeň odbornej klasifikácie: Inžinier v programe Informačné systémy. Žilina: FRI ŽU v Žiline, 2010. – 55 s.

Cieľom práce je návrh a tvorba webovej aplikácie, ktorá by umožňovala používateľom vzdialený prístup k simulátoru počítačových sietí Dynamips a tým dávala možnosť pracovať s reálnymi sieťovými topológiami.

## **Abstract**

ŠANKO, Martin: Web application for the student learning support as a capabilities extension of the Dynamips and Dynagen tools [Diploma thesis] – University of Žilina in Žilina. Faculty of management science and informatics; Department of Information and Communication Networks. – Supervisor of diploma thesis: Ing. Pavel Segeč, PhD – Academic degree: Engineer (Master) in Information Systems. Žilina: FRI ŽU in Žilina, 2010. - 55 p.

Aim of work is to create a web application that allows users remote access to computer networks simulator Dynamips and so giving the opportunity to work with real network topology.

**Pod'akovanie :**

Rád by som poďakoval môjmu vedúcemu diplomovej práce Ing. Pavlovi Segečovi PhD. za odborné rady pri tvorbe tejto diplomovej práce a pomoc pri riešení niektorých problémov.

## Obsah:

<b>Zoznam obrázkov:</b> .....	<b>8</b>
<b>Úvod</b> .....	<b>9</b>
<b>1 Zber a analýza požiadaviek</b> .....	<b>10</b>
1.1 Dostupné riešenia.....	10
1.1.1 Packet Tracer.....	10
1.1.2 Boson NetSim .....	11
1.1.3 Dynamips/Dynagen/GNS3 .....	12
1.1.4 Vyhodnotenie.....	13
1.2 Analýza požiadaviek na systém z hľadiska používateľov .....	14
1.3 Špecifikácia požiadaviek na aplikáciu .....	15
1.4 Analýza správania a možnosti nástroja Dynamips / Dynagen..	17
1.4.1 Dynamips .....	17
1.4.2 Dynagen.....	18
1.4.3 Spustenie jednoduchšej konfigurácie.....	21
1.4.4 Optimalizačné funkcie.....	23
1.4.5 Komunikácia s reálnou sieťou.....	26
1.4.6 Odchytávanie prevádzky .....	26
<b>2 Návrh riešenia</b> .....	<b>28</b>
2.1 Porovnanie technológií.....	28
2.2 Základný návrh .....	29
2.3 Webové rozhranie a jeho funkcie .....	30
2.3.1 Prihlásenie a registrácia .....	32
2.3.2 Správa užívateľov.....	34
2.3.3 Tvorba, editácia a mazanie topológií .....	35
2.3.4 Rezervovanie a správa rezervácií topológií.....	36
2.3.5 Spúšťanie a práca s topológiou .....	37
2.4 Databáza .....	39
2.4.1 Popis tabuliek .....	40
<b>3 Drupal</b> .....	<b>43</b>
<b>4 Implementácia</b> .....	<b>45</b>
4.1 Úvod do problematiky.....	45

4.2	Základný koncept .....	45
4.3	Vytvorenie modulu .....	46
4.3.1	Vytvorenie súborov modulu.....	46
4.3.2	Vytvorenie oprávnení .....	47
4.3.3	Vytvorenie položiek menu.....	47
4.3.4	Definovanie bloku .....	49
	<b>Záver .....</b>	<b>51</b>
	<b>Zoznam použitých zdrojov .....</b>	<b>52</b>
	<b>Zoznam skratiek .....</b>	<b>53</b>
<b>5</b>	<b>Príloha .....</b>	<b>54</b>
5.1	Správne písanie konfiguračného súboru .....	54
5.2	Ukážka webového rozhrania.....	55

## Zoznam obrázkov:

Obr. 1-1 Ukážka rozhrania Cisco Packet Traceru.....	11
Obr. 1-2 Boson NetSim .....	12
Obr. 1-3 Ukážka prostredia GNS3 .....	13
Obr. 1-4 Diagram prípadov použitia požiadaviek na Web rozhrania	16
Obr. 1-5 Diagram členenia konfiguračného súboru .....	19
Obr. 1-6 Schéma zapojenia .....	19
Obr. 1-7 Dynagen rozhranie s výpisom všetkých príkazov .....	21
Obr. 1-8 Spustený Dynamips server.....	22
Obr. 1-9 Spustený Dynagen spolu s výpisom zariadení.....	22
Obr. 1-10 Potenciálne hodnoty Idle-PC získané analýzou .....	23
Obr. 2-1 Návrh riešenia aplikácie .....	29
Obr. 2-2 Diagram aktivít pre prihlasovanie a registráciu .....	33
Obr. 2-3 Diagram aktivít pre správu užívateľov .....	34
Obr. 2-4 Diagram aktivít správy topológií.....	36
Obr. 2-5 Diagram aktivít pre rezerváciu.....	37
Obr. 2-6 Diagram aktivít pre spustenie topológie .....	38
Obr. 2-7 Databázový model.....	39
Obr. 5-1 Ukážka webovej aplikácie .....	55



## Úvod

V súčasnej dobe sa zvyšujú nároky na vzdelanostnú úroveň človeka z hľadiska nielen teoretických, ale najmä praktických vedomostí. Vzniká tak potreba zabezpečenia takých zdrojov poskytujúcich jednoduché možnosti prístupu k informáciám a interaktívnym prvkom, ktoré by mali dopomôcť k lepšiemu pochopeniu danej tematiky. V tomto smere sú veľmi užitočnými prostriedkami vzdelávania nové informačné technológie, vďaka ktorým sa vzdelávací proces stáva efektívnejší, ako s využitím bežných spôsobov výučby.

Medzi najrozšírenejšie patrí využívanie internetu a rôznych on-line kurzov, ktoré však užívateľovi pomáhajú iba pri získavaní teoretických znalostí. Získavanie praktických znalostí býva riešené formou reálnych kurzov. Tu však vznikajú rôzne obmedzenia, či už časové, personálne, alebo hardvérové. Táto práca sa zaoberá tvorbou takého nástroja, ktorý by riešil práve problém obmedzeného prístupu k zariadeniam, na ktorých by si mohli študenti reálne skúšať topológie a tým si zlepšovali praktické vedomosti z oblasti počítačových sietí.

# 1 Zber a analýza požiadaviek

Katedra informačných sietí na Fakulte riadenia a informatiky Žilinskej univerzity zabezpečuje výučbu predmetov zameraných najmä na sieťové technológie. Keďže sa na katedre nachádza päť laboratórií, z čoho iba dve sú vybavené sieťovými zariadeniami, vzniká iba obmedzený priestor pre študentov, ako sa dostať k zariadeniam.

Hlavným cieľom práce je vytvorenie webového rozhrania, ktoré by bolo schopné komunikovať s aplikáciou Dynamips, ktorá slúži na virtualizáciu hardvéru smerovačov firmy Cisco. Rozhranie by malo umožňovať vzdialený prístup študentom a učiteľom k im dostupným topológiám a prácu s nimi, zároveň by poskytovalo možnosť si jednotlivé topológie objednávať, spravovať užívateľov a mnoho ďalších funkcií.

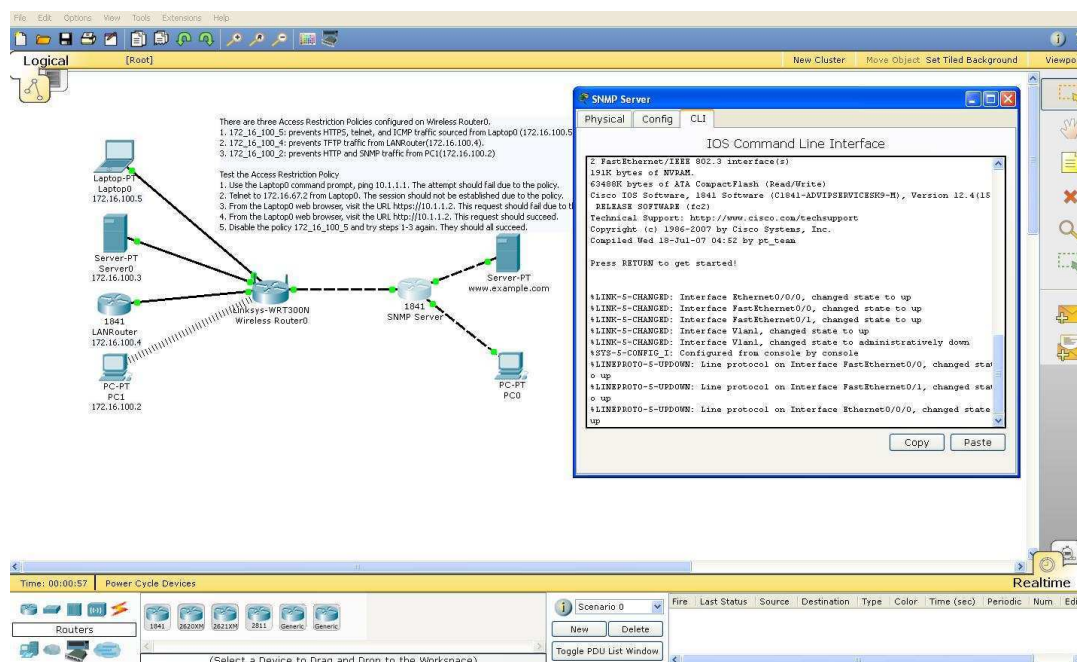
## 1.1 Dostupné riešenia

K dispozícií je dostupných viacero aplikácií, na ktorých by si študent mohol preveriť svoje vedomosti a praktické zručnosti. Keďže jednotlivé nástroje sa líšia najmä poskytovanou funkcionalitou, od najjednoduchších ponúkajúcich iba veľmi obmedzené možnosti až po tie komplexné, bolo by vhodné spomenúť iba najrozšírenejšie riešenia poskytujúce čo najväčšiu funkcionalitu.

### 1.1.1 Packet Tracer

Packet Tracer od spoločnosti Cisco je pravdepodobne najznámejším nástrojom na simuláciu sietí. Je to simulačný program, ktorý dovoľuje tvorbu ľubovoľných topológií a následnú prácu

s nimi. Výhodou tohto programu je, že je voľne dostupný pre študentov fakúlt zapojených do programu Cisco Academy[1].



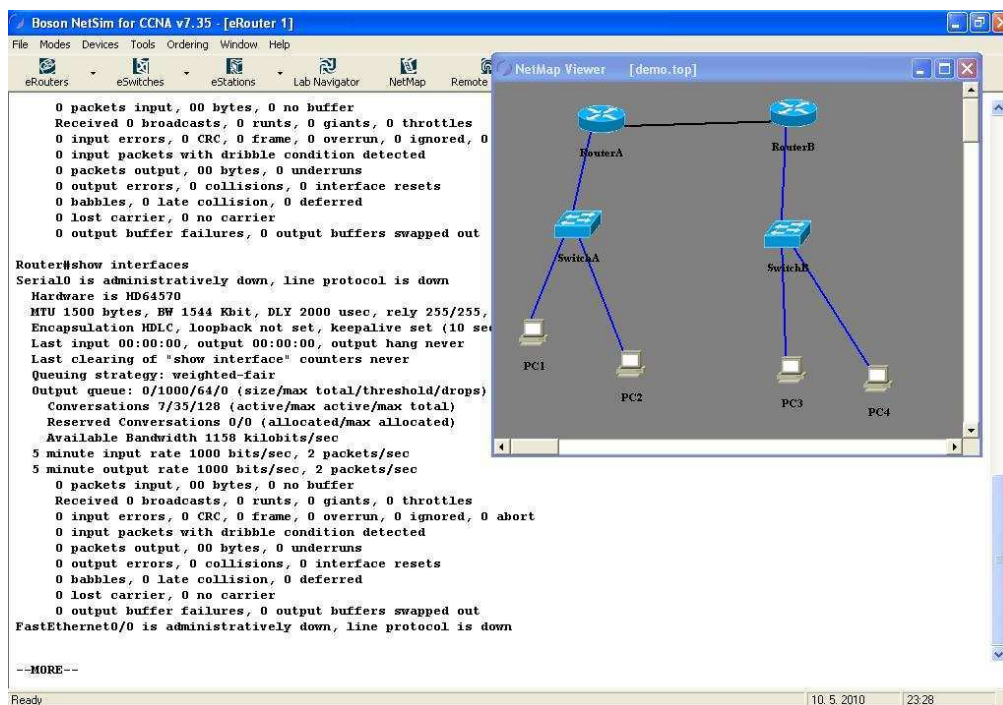
Obr. 1-1 Ukážka rozhrania Cisco Packet Traceru

PT vyniká ľahkým a intuitívnym ovládaním, krátkym časom na zhotovenie topológií, ich následným testovaním a konfigurovaním pomocou užívateľsky príjemného grafického prostredia. Nevýhodou tejto aplikácie je však jej obmedzená funkcionálnosť. Tá je zapríčinená tým, že program nevyužíva skutočný obraz operačného systému smerovača (IOS – Internetwork Operating System), ale iba funkcie implementované programom.

### 1.1.2 Boson NetSim

Boson NetSim[2] je simulačný program podobný PT. Program sa zameriava na simuláciu činnosti zariadení spoločnosti Cisco. Primárne je tento simulátor určený pre záujemcov o CCNA a CCNP certifikáciu. Program sa vyznačuje množstvom predpripravených sieťových topológií, ktoré môže užívateľ riešiť, alebo si môže

vytvárať vlastné zapojenia využitím 47 rozdielnych zariadení. Taktiež obsahuje návody a postupy riešení týchto topológií. Nevýhodou je ako aj u PT to, že sa jedná iba o simulátor, čiže neponúka plnú funkcionálnosť zariadení. Zároveň zaostáva svojou vizuálnou stránkou, a ovládateľnosťou za PT.



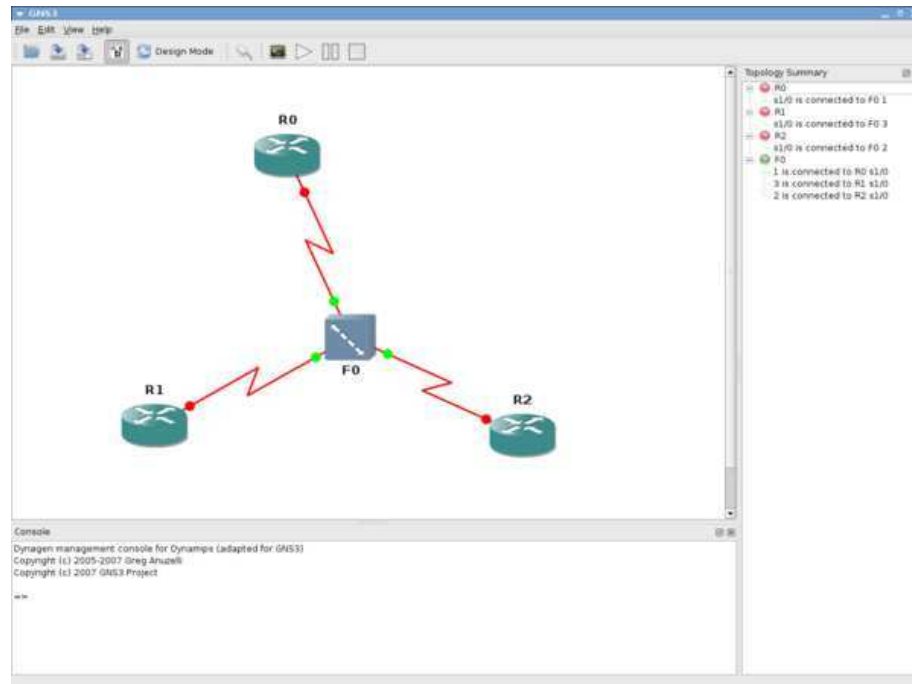
Obr. 1-2 Boson NetSim

### 1.1.3 Dynamips/Dynagen/GNS3

Ďalším nástrojom je voľne dostupný emulátor Cisco smerovačov Dynamips a jeho rozšírenie Dynagen, ktorých podrobnejší popis bude v ďalšej kapitole. Hlavným rozdielom medzi týmito nástrojmi je, že Dynamips na rozdiel od predošlých nástrojov, funkcie smerovačov nesimuluje, ale priamo emuluje pomocou Cisco IOSu reálnych zariadení. Nevýhodou je však potreba vlastniť zakúpený IOS k zariadeniam, ktoré by študent chcel používať, keďže sa nejedná o voľne dostupný softvér.

Vhodnou voľbou je použitie Graphical Network Simulator – GNS3, ktorý je grafickou nadstavbou Dynamipsu a Dynagenu. Jeho

hlavnou výhodou je, že spája užívateľsky príjemne prostredie a vzhľad pripomínajúci Packet Tracer s možnosťami ponúkanými Dynamipsom.



Obr. 1-3 Ukážka prostredia GNS3

### 1.1.4 Vyhodnotenie

Porovnaním jednotlivých programov a funkcionalít, ktoré poskytujú bolo možné určiť silné a slabé stránky daných aplikácií a z nich nakoniec vybrať tú, ktorá by najviac vyhovovala požiadavkám. Je ňou aplikácia Dynamips. Dôvodom jej výberu je najmä to, že na rozdiel od zvyšných riešení sa jedná o emulátor, čiže poskytne funkcionalitu reálnych zariadení. Zároveň je platformovo nezávislý a je schopný fungovať ako klient/server aplikácia. Je však náročný na výkon počítača.

Preto pre potreby katedry bola navrhnutá webová aplikácia, pomocou ktorej by mali študenti možnosť prístupu k nástroju Dynamips, umiestnenom na školskom serveri, v ktoromkoľvek čase a z ktoréhokoľvek miesta s prístupom na internet.

## 1.2 Analýza požiadaviek na systém z hľadiska používateľov

Z hľadiska rôznorodosti požiadaviek na systém a jeho využívanie je možné definovať 2 skupiny užívateľov:

- 1) Administrátor
- 2) Používatelia

Skupina *Administrátor*, ako už jej názov napovedá, popisuje užívateľa, ktorý by celý systém spravoval. Na rozdiel od skupiny *Používatel'ov* by systém nepoužíval, ale zabezpečoval jeho funkcionality a správu pre bezproblémový chod aplikácie.

Jeho základné úlohy by boli nasledovné:

- správa prvkov webu,
- správa obsahu,
- konfigurácia stránky,
- správa užívateľov a oprávnení,
- správa systémových prostriedkov a súborového systému,
- riešenie a odstraňovanie problémov.

Druhou skupinou je skupina *Používatelia*. Rozumieme pod ňou užívateľov, ktorí budú reálne využívať prostriedky aplikácie. Je možné sem zaradiť učiteľov a študentov. Práve pre túto skupinu je systém primárne vyvíjaný.

Ako hlavné požiadavky pre celú skupinu je možné považovať:

- tvorba topológií, ich editácia a mazanie,
- možnosť spúšťať a zastavovať vlastné topológie,
- možnosť rezervovať si danú topológiu na určitý deň a hodinu v týždni,
- zabezpečiť určitý počet hodín pre prácu s topológiami každému užívateľovi.

### **1.3 Špecifikácia požiadaviek na aplikáciu**

Medzi všeobecné požiadavky, ktoré by mala aplikácia spĺňať môžeme zaradiť:

- Prístupnosť aplikácie z internetu.
- Bezpečnosť.
- Jednoduché, užívateľsky príjemné ovládanie.
- Rozšíriteľnosť aplikácie o nové prvky.

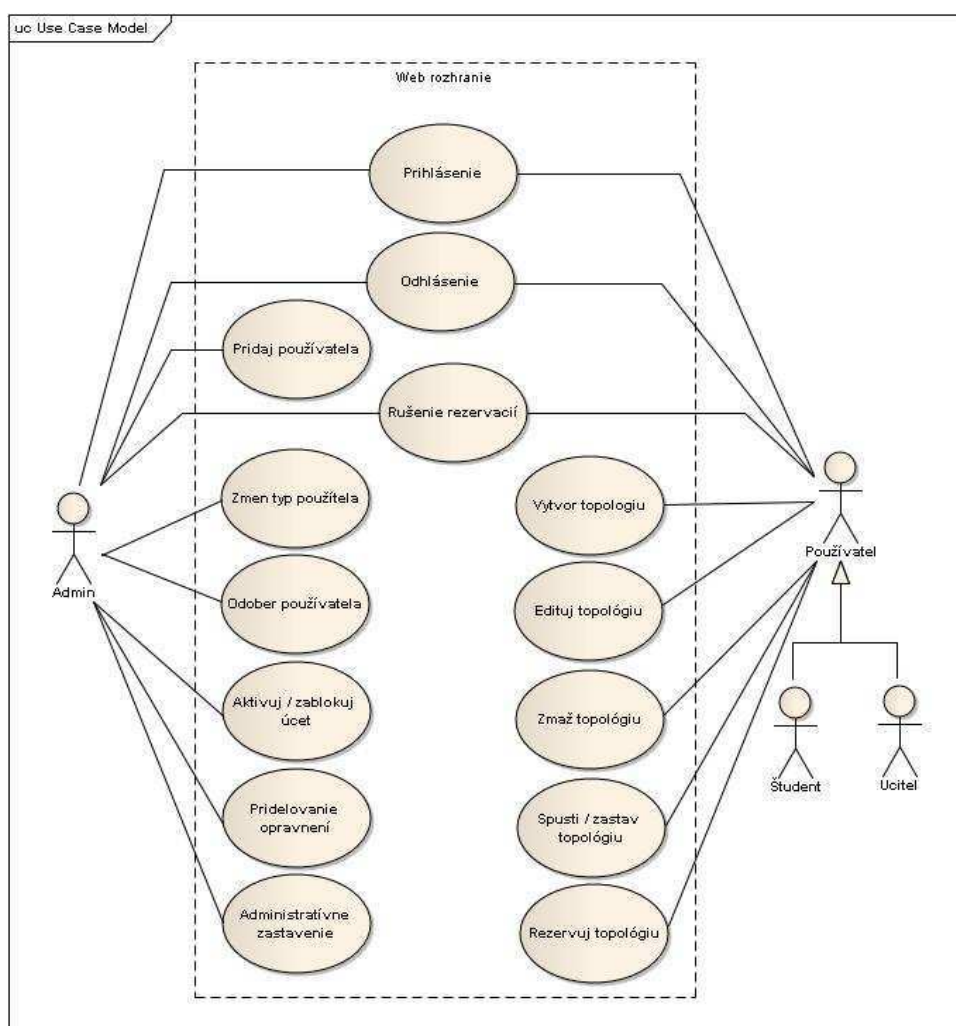
Z analýzy vyplýva, že celý systém by mal užívateľom poskytovať nasledovné funkcionality:

- Možnosť prihlásiť sa iba registrovaným užívateľom, aby sa zabránilo používaniu aplikácie nepovolaným osobám.
- Správa používateľských účtov – vytváranie, rušenie, blokovanie účtov, vytváranie používateľských skupín a zaraďovanie užívateľov do nich, tvorba prístupových pravidiel.
- Pridelovanie oprávnení na vykonávanie jednotlivých činností v rámci aplikácie(napr. tvorba topológií a podobne).
- Tvorba nových topológií a využívanie existujúcich.
- Editovať existujúcu topológiu.
- Zmazať existujúcu topológiu.
- Ukladanie rozpracovaných topológií a možnosť v pokračovaní konfigurácie neskôr.
- Rezervovať si prístupnú topológiu na vybraný časový blok v rámci týždňa s garanciou maximálnej dĺžky rezervácie.
- Zrušiť rezerváciu.
- Administrácia rezervácií.
- Spustiť / zastaviť objednanú topológiu.
- Spustiť topológiu aj bez objednania ak má server ešte voľné kapacity.
- Spravovať systémové prostriedky.

- Sledovanie systému.

Jednotlivé skupiny užívateľov by mali byť schopné využívať vyššie uvedené funkcionality v závislosti od oprávnení definovaných pre ich konkrétnu skupinu, ktorej členom sú.

Na základe špecifikácie požiadaviek na aplikáciu bol zhotovený nasledovný diagram prípadov použitia požadovaných funkcií, ktorý by mal prehľadne ilustrovať, čo všetko bude aplikácia z hľadiska funkcionality ponúkať.



**Obr. 1-4 Diagram prípadov použitia požiadaviek na Web rozhrania**

Z obrázku vidno, že pôvodná skupina *Používatelia* bola rozčlenená na dve podskupiny:



- študenti,
- učitelia

a to z dôvodu jednoduchšej správy oprávnení prístupu k jednotlivým prvkom webového rozhrania.

## **1.4 Analýza správania a možnosti nástroja Dynamips / Dynagen**

### **1.4.1 Dynamips**

Dynamips je multiplatformový emulátor procesoru MIPS (MIPS - Microprocessor without Interlocked Pipeline Stages) a hardvérového prostredia, naprogramovaný Christopherom Fillotom, pôvodne za účelom emulácie Cisco smerovačov rady 7200 na klasickom počítači. Jeho vývojom a rozširovaním dospel až do dnešnej podoby, kedy okrem rady 7200 podporuje série smerovačov 3600 (modely 3620, 3640 a 3660), série 3700 (modely 3725, 3745) a sériu 2600 (modely 2610, 2650XM, 2691).

#### **Princíp činnosti**

Aplikácia si po spustení vytvorí virtuálne prostredie, ktoré bude emulovať skutočné zariadenie zvoleného typu. V tomto prostredí je potom možné spúšťať reálny operačný systém. Týmto má užívateľ k dispozícii úplnú funkcionálnu reálneho zariadenia. Táto funkcionálna je užívateľovi sprístupnená pomocou konzoly. Použitím nastavení je možné navyše prepojiť sieťovú kartu počítača s fast-ethernet portom zariadenia a tým ho pripojiť do živej siete. Okrem možnosti pripojenia na reálnu topológiu je možnosť pripojiť sa na iné virtuálne zariadenie, prípadne viacero zariadení a tým vytvoriť prakticky akúkoľvek virtuálnu topológiu.

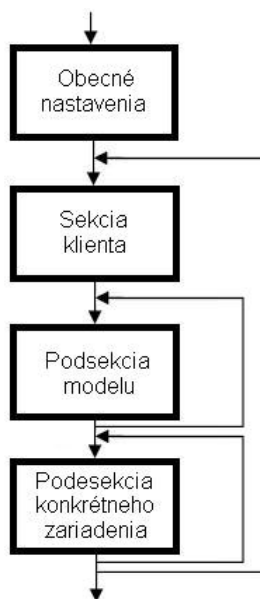
Dynamips vyžíva značné množstvo zdrojov počítača v podobe operačnej pamäte (RAM – Random Acces Memory) a vyťaženia procesora (CPU – Central Processing Unit). Pre spustenie obrazu operačného systému smerovača, ktorý by na reálnom smerovači vyžadoval 256MB RAM, si aplikácia vyhradí v pamäti rovnako veľké miesto aj pri jeho virtuálnej verzií. Okrem toho si vyhradí ďalších 64MB RAM v prípade inštancie spustenej na UNIX platforme alebo 16MB v prípade Windows pre preklady cache JIT (JIT – Just in time).

Keďže program emuluje CPU smerovača a vykonávanie všetkých jeho inštrukcií, značne vyťažuje CPU počítača, na ktorom beží inštancia Dynamipsu. Je to spôsobené tým, že CPU počítača nerozlišuje medzi stavom kedy sa vykonávajú inštrukcie potrebné pre prácu a stavom kedy sú smerovačom vykonávané procesy v nečinnosti.

Na riešenie týchto nedostatkov sa využívajú optimalizačné funkcie, ktoré budú popísané neskôr, keďže súvisia zároveň aj s Dynagenom.

### **1.4.2 Dynagen**

Dynagen je nástroj používaný v kombinácii s aplikáciou Dynamips. Jedná sa o nadstavbu vo forme konzolového rozhrania, využívajúcu Hypervisor mód na komunikáciu s Dynamipsom. Jeho hlavnou úlohou je zjednodušiť vytváranie a prácu s topológiami a to tak, že ich vytváranie prebieha v rámci konfiguračného súboru. Jedná sa o textový súbor s príponou .net v ktorom sú špecifikované zariadenia a prepojenia medzi nimi.



**Obr. 1-5 Diagram členenia konfiguračného súboru**

Ako príklad takéhoto konfiguračného súboru uvidíme a následne vysvetlíme jednoduché prepojenie dvoch smerovačov modelu 3620, prepojených pomocou ich sériových portov S1/0, ktoré je znázornené na obrázku.



**Obr. 1-6 Schéma zapojenia**

```
# Simple lab

ghostios = True
sparsemem = True

[localhost]

[[3620]]
    image = /opt/7200-images/c3620-i-mz.123-22.bin.unp
    ram = 128

[[ROUTER R1]]
    s1/0 = R2 s1/0
    console = 2000

[[ROUTER R2]]
```

```
console = 2001
```

Ako je vidno, súbor má pomerne jednoduchú syntax a štruktúru. Prvú sekciu tvoria obecné nastavenia. V nich je možné definovať nastavenia pre celý konfiguračný súbor. Ďalšou časťou je sekcia klienta. Tvorí ju definovanie adresy kde beží Dynamips v tvare [<hostname>:<port>].

Parameter hostname môže mať nasledujúce hodnoty:

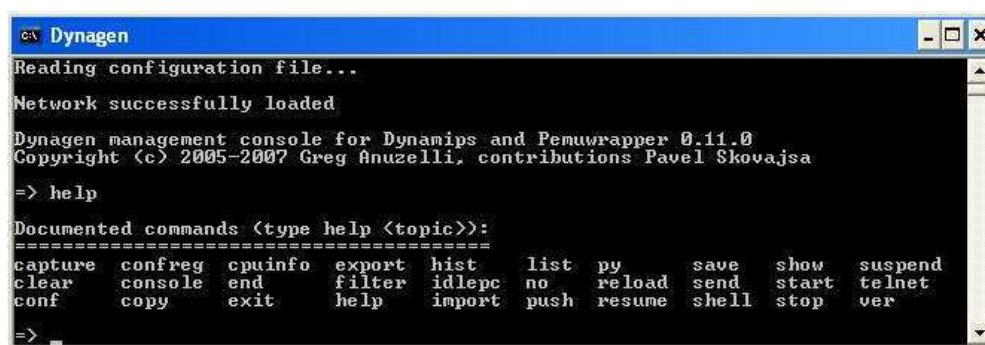
- Localhost – v prípade že dynamips server a dynagen bežia na rovnakom PC,
- IP adresa,
- doménové meno definované s súbore host.txt.

Ďalšia sekcia je podsekcia modelu. Sú v nej definované všetky typy modelov zariadení využívaných klientom definovanom v sekcii klienta a ich nastavenia. V tomto prípade [[3620]] znamená, že nasledujúca konfigurácia sa vzťahuje na Dynamips server bežiaci na localhoste. Zároveň udáva nastavenia, ktoré budú použité pre všetky inštancie smerovača konkrétneho modelu (v tomto prípade 3620) a nie je potrebné ich udávať pri každom zariadení osobitne. Položka image udáva cestu k IOS pre dané zariadenie. RAM definuje, koľko MB RAM bude vyhradených pre každú inštanciu smerovača daného typu.

Poslednou sekciou je podsekcia konkrétneho zariadenia. Jednotlivé inštancie smerovačov sú definované pomocou kľúčového slova ROUTER a názvom napr. [[ROUTER R1]]. Názov je využívaný iba v rámci Dynagenu, čiže sa nejedná o hostname daného zariadenia v topológii. Potom nasleduje definovanie prepojenia jednotlivých smerovačov a to jednoduchým zápisom napr. s1/0 = R2 s1/0. V prípade potreby prepojenia rozhrania s reálnym rozhraním je tu možnosť prepojenia definovaním názvu tohto rozhrania. Bližší popis toho bude spomenutý v podkapitole Správanie a možnosti Dynamips a dynagen. Ďalej je možné definovať konzolové porty pomocou ktorých bude realizované telnetové pripojenie na dané zariadenia.

Okrem týchto najzákladnejších funkcií, ktoré boli popísané iba kvôli lepšiemu pochopeniu štruktúry konfiguračného súboru, je možnosť využitia mnohých iných príkazov, ktoré sú uvedené v tabuľkách v prílohe.

Dynagen obsahuje CLI rozhranie – Manažment konzolu pomocou ktorej je možné vypísať zoznam zariadení, spúšťať, zastavovať, reštartovať, pozastavovať a obnovovať činnosť virtuálnych smerovačov použitých v danej topológii, odchytať prevádzku a mnoho iných funkcií.



```

c:\ Dynagen
Reading configuration file...
Network successfully loaded

Dynagen management console for Dynamips and Pemuwrapper 0.11.0
Copyright (c) 2005-2007 Greg Anuzelli, contributions Pavel Skovajsa

=> help

Documented commands (type help <topic>):
=====
capture  confreg  cpuinfo  export  hist    list  py    save  show  suspend
clear    console  end      filter  idlepc  no    reload send  start telnet
conf     copy    exit     help    import  push  resume shell stop  ver

=>

```

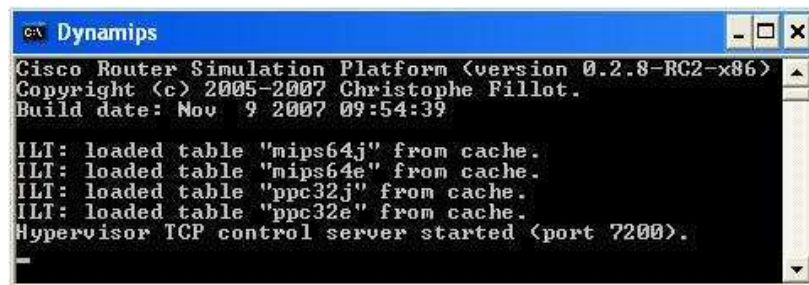
Obr. 1-7 Dynagen rozhranie s výpisom všetkých príkazov

Zároveň je možné sa z tohto rozhrania pripojiť na konzolové porty jednotlivých zariadení. Predtým je ale potrebné nastaviť v konfiguračnom súbore programu dynagen.ini, ktorý telnetový klient sa má použiť. Nastavenie prebieha jednoduchým odkomentovaním konkrétneho riadku v rámci konfiguračného súboru.

### 1.4.3 Spustenie jednoduchkej konfigurácie

V závislosti od operačného systému na ktorom má Dynamips server bežať, sa líši aj spôsob jeho spúšťania. V prípade Windows je možné spúšťať server pomocou odkazu na ploche, ktorý sa vytvorí po inštalácii, alebo cez príkazový riadok.

Na systémoch Linux/Mac/UNIX sa server spúšťa pomocou príkazu, napr. `dynamips -H 7200`. To spustí server v Hypervisor móde, ktorý počúva na definovanom porte (v tomto prípade 7200).



```
C:\> Dynamips
Cisco Router Simulation Platform (version 0.2.8-RC2-x86)
Copyright (c) 2005-2007 Christophe Fillot.
Build date: Nov  9 2007 09:54:39

ILT: loaded table "mips64j" from cache.
ILT: loaded table "mips64e" from cache.
ILT: loaded table "ppc32j" from cache.
ILT: loaded table "ppc32e" from cache.
Hypervisor TCP control server started (port 7200).
```

Obr. 1-8 Spustený Dynamips server

Po spustení dynamips servera je potrebné spustiť danú topológiu. V OS Windows stačí iba spustiť priamo súbor s topológiou, čo spustí aj Dynagen. V Linuxe/Unixe je potrebné .net súbory asociovať s Dynagenom, alebo spúšťať pomocou príkazového riadku príkazom `dynagen nazov.net` (v tomto prípade `dynagen simple1.net`).



```
C:\> Dynagen
Reading configuration file...
Network successfully loaded

Dynagen management console for Dynamips and Pemuwrapper 0.11.0
Copyright (c) 2005-2007 Greg Anuzelli, contributions Pavel Skovajsa

=> list

```

Name	Type	State	Server	Console
R1	3620	running	localhost:7200	2000
R2	3620	running	localhost:7200	2001

```
=>
```

Obr. 1-9 Spustený Dynagen spolu s výpisom zariadení

Po spustení sa vykoná syntaktická a logická kontrola správnosti spúšťanej topológie. Ak neobsahuje chyby vytvoria sa jednotlivé inštancie zariadení a prepojenia medzi nimi a užívateľovi sa sprístupní Manažment konzola. V prípade chyby je užívateľ informovaný chybovou správou.

## 1.4.4 Optimalizačné funkcie

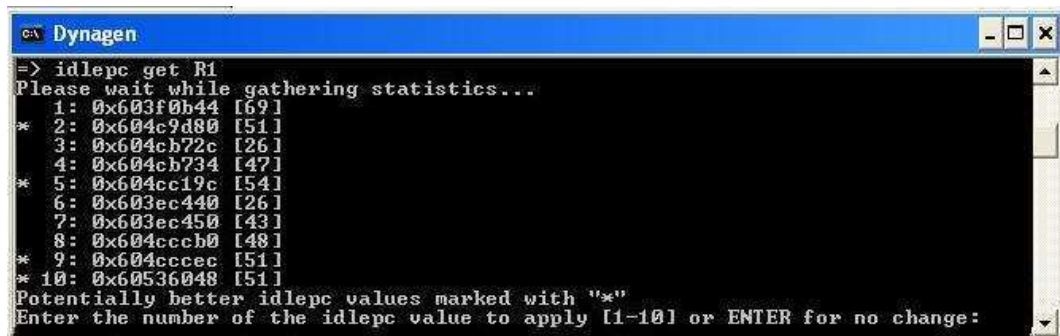
Ako bolo už spomenuté, Dynamips značne vyťažuje PC a preto sa využívajú optimalizačné funkcie ktoré sa tento problém snažia riešiť.

### Idle-PC

Najvýraznejšou z nich je funkcia Idle-PC, ktorá rieši problém 100% vyťažovania CPU počítača, na ktorom beží inštancia dynamips servera. Táto funkcia vykonáva analýzu v rámci spusteného imagu súboru a snaží sa určiť bod vo vykonávanom kóde, ktorý reprezentuje nečinnosť IOSu.

Proces zistenia hodnoty Idle-PC pozostáva zo spustenia topológie. Nasleduje spustenie smerovača pre ktorý chceme hodnotu Idle-PC zistiť. Po spustení nasleduje pripojenie na dané zariadenie pomocou telnetu. Zariadenie nabootuje IOS. V rozhraní dynagenu je potrebné spustiť príkaz *idlepc get nazov\_smerovaca*.

Program začne analyzovať bežiaci IOS a po chvíli ponúkne užívateľovi zoznam potenciálnych hodnôt Idle-PC ako je vidno na nasledovnom obrázku.



```
C:\> Dynagen
=> idlepc get R1
Please wait while gathering statistics...
1: 0x603f0b44 [69]
* 2: 0x604c9d80 [51]
3: 0x604cb72c [26]
4: 0x604cb734 [47]
* 5: 0x604cc19c [54]
6: 0x603ec440 [26]
7: 0x603ec450 [43]
8: 0x604cccb0 [48]
* 9: 0x604cccec [51]
* 10: 0x60536048 [51]
Potentially better idlepc values marked with "*"
Enter the number of the idlepc value to apply [1-10] or ENTER for no change:
```

Obr. 1-10 Potenciálne hodnoty Idle-PC získané analýzou

Najpravdepodobnejšie hodnoty sú v zozname označené \*. Po zvolení danej hodnoty by malo vyťaženie procesora rapídne klesnúť.

Ak vyťaženie procesora nekleslo, je možné príkazom *idlepc show nazov\_smerovaca* opätovne zobrazíť zoznam potenciálnych hodnôt a zvoliť inú, až pokiaľ nie je nájdená správna hodnota.

Tieto hodnoty je možné následne uložiť do databázy vo forme súboru s názvom *dynagenidledb.ini*, alebo priamo do .net súboru s topológiou a pri ďalšom spúšťaní dynagenu budú automaticky načítané a použité. V prípade použitia databázy nie je ukladaná hodnota pre konkrétne zariadenie ale pre IOS image. V závislosti od verzie IOS sa mení aj hodnota Idle-PC. Tá však nie je závislá od operačného systému, hostiteľského PC, ani od verzie dynamipsu čiže hodnoty databázového súboru je možné prenášať, kopírovať alebo pridávať z iných súborov. Je potrebné však zachovať unikátnosť názvov IOS súborov, nakoľko je databáza indexovaná pomocou názvov.

### **Ghost IOS**

Ghost IOS je optimalizačná funkcia, ktorá dokáže výrazne zredukovať množstvo využívanej pamäte v počítači v prípade spúšťania topológie, v ktorej sa nachádza viacej inštancií zariadenia využívajúceho rovnaký IOS. Podstata tejto funkcie spočíva v tom, že miesto toho aby si každé zariadenie vytváralo a uchovávalo identickú kópiu IOS vo svojej virtuálnej pamäti, je alokovaná jedna zdieľaná oblasť pamäti, ktorú budú využívať všetky zariadenia. Táto funkcia má opodstatnenie iba pri topológiach s viacerými inštanciami smerovačov využívajúcimi rovnaký IOS. Čím viacej takýchto zariadení typológia obsahuje, tým je množstvo ušetrenej pamäte väčšie.

Pre povolenie tejto funkcie stačí použiť príkaz *ghostios = true* v konfiguračnom súbore topológie. Toto nastavenie je možné použiť buď na najvyššej úrovni, kedy je potom platné pre všetky inštancie smerovačov, alebo na úrovni definície jednotlivých typov čo má za následok aplikáciu nastavení iba pre daný model.

### **Sparsemem**

Hlavnou úlohou *sparsemem* je redukovanie množstva virtuálnej pamäte využívanej jednotlivými inštanciami smerovačov. Je to dôležité najmä z hľadiska, že operačný systém PC limituje veľkosť



jedného procesu na 2GB virtuálnej pamäte v prípade 32 bitovej verzie Windows a na 3GB pre Linux. Týmto vznikajú obmedzenia v podobe maximálneho množstva spustených inštancií smerovačov.

Princípom funkcie je alokácia iba takého množstva virtuálnej pamäte na hosťovskom PC aké je vyžadované IOS-om daného smerovača. Toto umožňuje spustenie viacerých inštancií smerovačov v rámci jednej inštancie dynamipsu bez potreby spúšťania ďalšej. Povoľovanie funkcie prebieha rovnako ako aj u predošlých funkcií, čiže jednoduchým pridaním *sparsemem = true* do najvyššej úrovne v rámci konfiguračného súboru topológie.

### 1.4.5 Komunikácia s reálnou sieťou

Ako bolo už spomenuté, dynamips dokáže prepojiť virtuálne rozhranie smerovača s reálnym, čo umožňuje prepojenie virtuálnej siete s reálnou sieťou a ich komunikáciu.

V závislosti od operačného systému sa líši aj spôsob tohto prepojenia. Na operačnom systéme Linux je ho možné dosiahnuť pomocou *NIO\_linux\_eth* *NIO* popisovača napr. *f0/0* = *NIO\_linux\_eth:eth0*. Týmto vznikne prepojenie medzi fast-ethernetovým rozhraním *f0/0* smerovača so sieťovou kartou v Linuxe identifikovanou ako *eth0*.

V prípade Windows, je prepojenie zabezpečované knižnicou Winpcap. Opäť je však potrebné vedieť ako je systémom identifikovaná sieťová karta. To sa dá jednoducho zistiť jednoduchým programom obsiahnutým v inštalačnom balíku. Nazýva sa Network device list a jeho jedinou funkciou je identifikácia sieťových kariet v PC. Po zistení hodnôt je realizované prepojenie použitím zápisu:

*F0/0* = *NIO\_gen\_eth:\Device\NPF\_{ }*.

Zápis v tomto by mohol vyzeráť nasledovne:

*F0/0* = *NIO\_gen\_eth:\Device\NPF\_{867A1EC3-0822-4163-A533-BFD1C8AA48D0}*

### 1.4.6 Odchytávanie prevádzky

Odchytávanie prevádzky je ďalšou z funkcionalít ktorú Dynamips/Dynagen ponúka. Je možné ju realizovať na sériových alebo Ethernetových rozhraniach virtuálnych smerovačov. Jedná sa o klasické odchytávanie paketov a ich následné ukladanie do súboru. Tento súbor je možné následne otvoriť v ktoromkoľvek programe schopnom čítať súbory typu *libpcap*.

Spustenie odchytávania paketov sa uskutočňuje príkazom z rozhrania dynamipsu zadáním príkazu *capture* definovaním

smerovača na ktorom chceme odchyťávať, konkrétneho rozhrania a súboru kam sa má prevádzka uložiť napr. :

*capture r1 f0/0 subor.cap*

pre zastavenie odchyťávania treba použiť rovnaký príkaz iba s tým rozdielom, že na začiatku sa doplní “no“ a z konca sa odoberie názov súboru napr. :

*no capture r1 f0/0*

V prípade odchyťávania prevádzky na sériových rozhraniach treba navyše špecifikovať typ zapúzdrenia (FR, HDLC, PPP) využívaného smerovačmi jeho pridaním za príkaz na odchyťávanie ako pri Ethernetových rozhraniach. Je to nevyhnutné pre správne dekodovanie paketov.

## 2 Návrh riešenia

Veľmi dôležitou časťou pri vývoji aplikácie je vypracovanie kvalitného návrhu, pretože do značnej miery ovplyvní celkovú funkcionálnosť systému, jeho rozšíriteľnosť a hlavne jeho životnosť.

### 2.1 Porovnanie technológií

Je dostupných mnoho technológií, ktoré by sa dali použiť pri implementácii riešenia tohto systému. Nakoľko by sa však malo jednať o riešenie dostupné z webu, najvhodnejšími riešeniami sú PHP a ASP.NET.

#### ASP.NET

Je moderný programovací framework pre vývoj webových aplikácií.

Medzi jeho hlavné výhody patrí:

- Rýchlosť aplikácií, keďže jeho kód je kompilovaný,
- rýchly vývoj aplikácií vďaka,
- množstvo programovacích jazykov.

Nevýhody:

- Vyššie systémové nároky.

#### PHP

PHP je populárny skriptovací jazyk na tvorbu dynamických webových stránok.

Výhody:

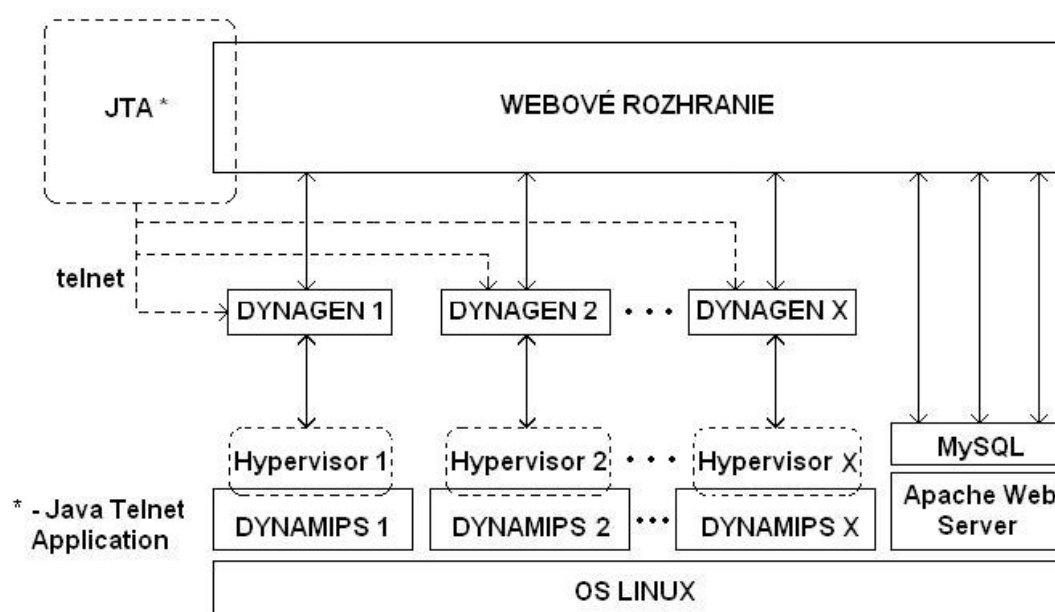
- Rýchlosť,
- nízke nároky na systém,
- platformová nezávislosť,
- veľmi rozšírené, veľká podpora, široká komunita,
- jednoduchý na naučenie sa.

Nevýhody:

- Menšia rýchlosť v porovnaní s ASP.NET.

## 2.2 Základný návrh

Po dôkladnom preskúmaní funkcionalít a možností nástroja Dynamips a Dynagen v predošlej kapitole a zohľadnení všetkých požiadaviek na aplikáciu, ktoré vyplývajú z analýzy bolo možné navrhnúť štruktúru aplikácie znázornenú nasledovným diagramom:



Obr. 2-1 Návrh riešenia aplikácie

Ako je z obrázku vidno, jedná sa o riešenie postavené na platforme operačného systému Linux. Z hľadiska pamäťovej náročnosti programu Dynamips je najvhodnejšie použiť 64 bitovú verziu z dôvodu možnosti použitia viac ako 4GB RAM.

Jadro aplikácie bude tvorené simulačným programom Dynamips, ktorý bude zabezpečovať emuláciu virtuálnych zariadení. Dynamips bude rozšírený nadstavbou Dynagen kvôli uľahčeniu spúšťania topológií.

Webové rozhranie by malo zabezpečovať komunikáciu medzi užívateľom a funkcionalitou poskytovanou serverom. Malo by mať podobu webovej stránky, z ktorej by mali mať užívatelia prístupné všetky možnosti aplikácie.

Pre ukladanie dát využívaných webovým rozhraním a aplikáciami Dynamips a Dynagen bude vytvorená databáza použitím databázového systému MySQL, s ktorou bude komunikovať webové rozhranie.

Riešenie komunikácie s jednotlivými virtuálnymi zariadeniami bolo v prvotnom návrhu plánované pomocou konzoly operačného systému, ale v záujme zachovania bezpečnosti aplikácie a zvýšenia pohodlnosti práce pre užívateľa bol návrh zmenený a bolo prijaté riešenie komunikácie pomocou java appletu – Java Telnet Application, prístupného priamo z webu.

Z hľadiska širokého spektra požiadaviek na aplikáciu, je vhodné použiť redakčný systém (CMS – Content Management System) poskytujúci základnú funkcionalitu ako prihlasovanie, správa užívateľov a podobne, a pri vývoji aplikácie sa sústrediť na jeho rozšírenie a doplnenie funkcionality vyžadovanej aplikáciou. Ako najvhodnejšie riešenie sa ukazuje použitie PHP CMS Drupal a vytvorenie modulu, ktorý by zabezpečoval komunikáciu s programami Dynamips a Dynagen. Podrobnejší popis CMS a jeho funkcionality bude uvedený v ďalšej kapitole.

## **2.3 Webové rozhranie a jeho funkcie**

Celý systém bude mať podobu webovej stránky, odkiaľ ho budú jeho užívatelia používať. Systém bude rozlišovať tri druhy užívateľov identifikovaných v kapitole 1.3:

- študent,
- učiteľ,
- administrátor.

Po prihlásení do systému sa užívateľovi sprístupnia možnosti, ktoré systém ponúka, v závislosti od oprávnení, ktoré má jeho skupina pridelené.

Užívateľ zo skupiny študent bude mať možnosť:

- Upraviť si svoj osobný profil (zmeniť si e-mail, heslo, osobné údaje a podobne).
- Prezerat' si zoznam jemu dostupných a rezervovaných topológií v rámci daného týždňa.
- Objednat' si ním zvolenú topológiu na určitý deň a konkrétny časový blok, ktorý má voľné kapacity.
- Možnosť zrušiť objednávku pred začatím rezervovaného termínu.
- Spustiť a pracovať so simuláciou v daný rezervovaný časový blok.
- Možnosť uložiť si prácu a možnosť sa k nej po určitom čase vrátiť a pokračovať.

Užívateľ zo skupiny učiteľ má prístupné všetky možnosti ako aj študent, ale jeho práva sú doplnené o ďalšie možnosti:

- Vytvoriť novú topológiu.
- Editovať, mazať, alebo sprístupňovať/zakazovať už existujúce topológie.
- Pridávať nových užívateľov.
- Editovať a mazať existujúcich užívateľov.
- Pridelovanie užívateľských práv užívateľom a zaradovanie do užívateľských skupín.
- Prezerat' si zoznam práve bežiacich simulácií a všetkých nadchádzajúcich rezervácií.
- Možnosť zrušiť rezerváciu užívateľovi z určitých dôvodov (napr. údržba systému), alebo zastaviť bežiacu simuláciu.

Posledný typ užívateľa je administrátor, ktorý má zo všetkých užívateľov najväčšie možnosti, ktoré mu aplikácia ponúka. Má

prístupné funkcionality predchádzajúcich dvoch skupín a navyše môže:

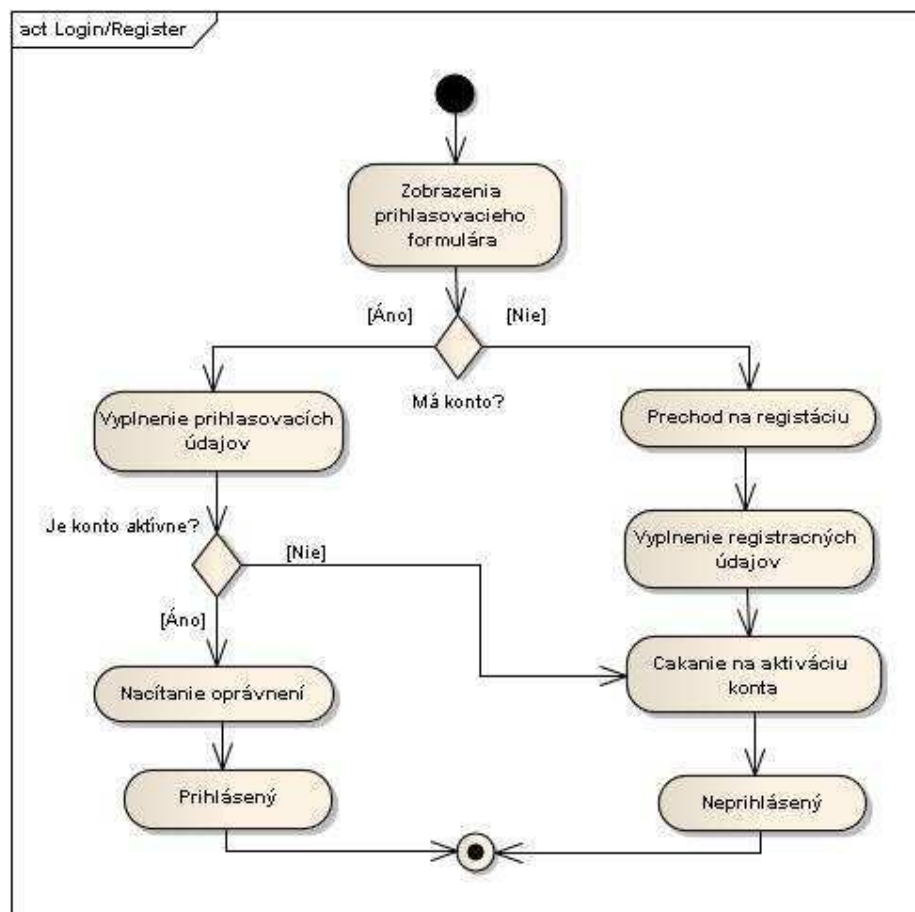
- Meniť vzhľad a obsah webu.
- Vykonávať údržbu.
- Meniť nastavenia systému (určovanie veľkosti časových blokov prístupných na rezerváciu, počet hodín prístupných študentovi na prácu v rámci týždňa).

### **2.3.1 Prihlásenie a registrácia**

Proces komunikácie užívateľa so systémom bude prebiehať iba pomocou webového rozhrania. Nato, aby užívateľ mohol so systémom pracovať, je potrebné aby sa do systému prihlásil. Prihlásenie je však podmienené tým, aby užívateľ bol v CMS systéme zaregistrovaný. Registrácia nových užívateľov môže prebiehať dvomi spôsobmi.

Prvým spôsobom je, že sa užívateľ zaregistruje sám pomocou formulára, ktorý bude na stránke prístupný. Tu vyplní všetky potrebné údaje, ktoré sú vyžadované systémom (užívateľské meno, e-mail, heslo, osobné informácie) a formulár potvrdí. Tým v systéme vznikne nový užívateľský účet, ktorý je však neaktívny a nie je pridelený do žiadnej užívateľskej skupiny, čiže nemá žiadne oprávnenia na prácu so systémom. Nato aby sa účet stal aktívny je potrebný administrátor, alebo učiteľ s dostatočnými právami na vytváranie nových užívateľských účtov. Schválením účtu sa zároveň odošle mail užívateľovi, ktorý žiadal o registráciu s informáciou, že jeho účet je aktívny a môže začať používať aplikáciu.





Obr. 2-2 Diagram aktivít pre prihlasovanie a registráciu

Druhý spôsob je, že účet vytvorí administrátor alebo učiteľ priamo zo svojej administračnej časti webového rozhrania. Postup je prakticky totožný s predchádzajúcim postupom len s výnimkou, že údaje o registrovanom užívateľovi do systému nezadáva osoba žiadajúci o registráciu.

Ak má užívateľ vytvorené konto a zároveň bolo aktivované administrátorom alebo učiteľom, môže sa užívateľ pomocou prihlasovacieho formulára zadaním užívateľského mena a hesla prihlásiť do systému a začať ho používať.

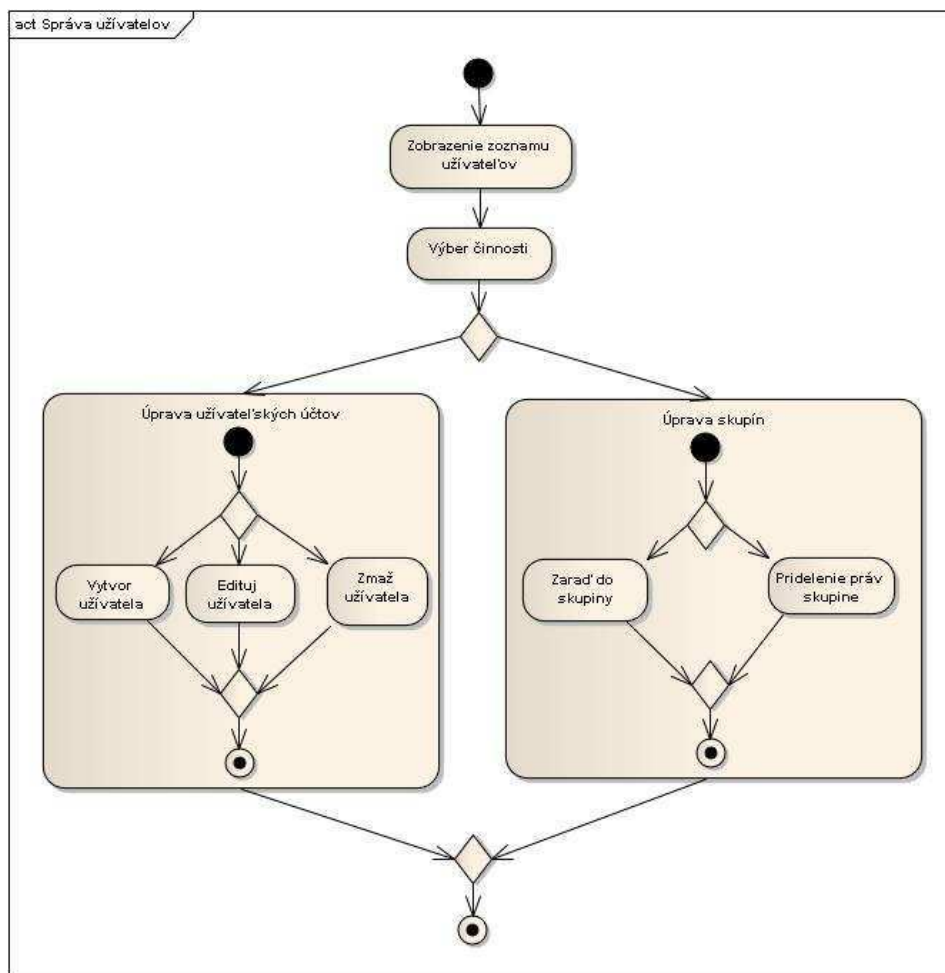
### 2.3.2 Správa užívateľov

Každý užívateľ zo skupiny učiteľov, alebo administrátor má vo svojej administračnej časti webového rozhrania prístupný zoznam všetkých užívateľov. V rámci neho môže užívateľské účty:

- vytvárať,
- upravovať,
- mazať,

a spravovať jednotlivé užívateľské skupiny tým, že:

- zaraďuje užívateľov do užívateľských skupín,
- meniť priradené oprávnenia jednotlivým skupinám.



Obr. 2-3 Diagram aktivít pre správu užívateľov

### 2.3.3 Tvorba, editácia a mazanie topológií

Procesu tvorby novej, editácie a mazania existujúcej topológie predchádza prihlásenie a potreba byť členom užívateľskej skupiny učiteľ alebo administrátor. Pre pridanie novej topológie, užívateľ použije položku menu *nová topológia*, kedy sa mu zobrazí formulár na vytváranie topológie. V ňom je potrebné zadať všetky povinné údaje:

- Názov (musí byť unikátny) pomocou ktorého bude topológia po svojom vytvorení v systéme identifikovaná,
- text konfiguračného súboru, ktorý musí mať štruktúru podľa predpisu popísaného v podkapitole Dynagen, pričom musí byť zachované, že konfiguračný súbor nesmie obsahovať žiadne konkrétne údaje o portoch, ale iba premenné predpísaného formátu (napr. consPort, hypervisorPort, udpPort),
- pamäťové nároky, ktoré udávajú celkovú pamäťovú náročnosť zariadení v topológií vyjadrenú v MB.

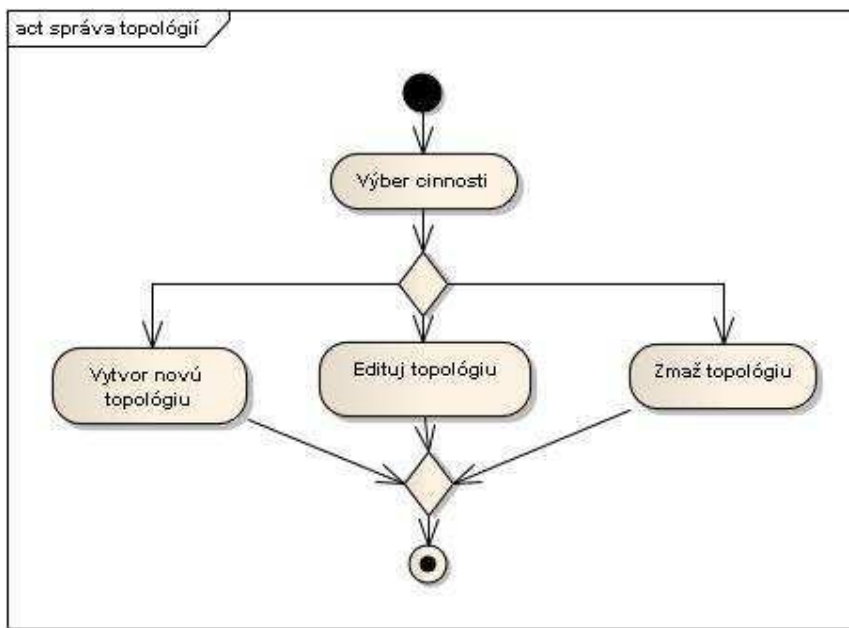
a voliteľné položky:

- Popis, ktorý by mal stručne charakterizovať danú topológiu, aby mal užívateľ informácie o aký typ topológie sa jedná,
- povolenie topológie, ktoré danú topológiu buď sprístupňuje alebo zakazuje pre študentov na objednávanie a spúšťanie,
- obrázok obsahujúci schému zapojenia danej topológie, ktorý sa bude zároveň zobrazovať užívateľovi pri spustení topológie pre ľahšiu orientáciu v jej zapojení.

Následne je možné topológiu uložiť do databázy.

V prípade procesu editácie alebo mazania je potrebné si najprv zobrazíť všetky dostupné topológie, ktoré sa v systéme nachádzajú. Z nich si vybrať tú, nad ktorou má byť vykonaná niektorá z uvedených operácií. Ak sa jedná o akciu editácie otvorí sa formulár s načítanými hodnotami takmer totožný s formulárom vytvárania novej topológie, iba s tým rozdielom že názov topológie nie je možné editovať. Užívateľ vykoná požadované zmeny a topológiu uloží do

databázy, alebo sa vráti na zoznam všetkých topológií, kde môže vykonávať zvyšné činnosti. Pri mazaní je zobrazený jednoduchý formulár informujúci užívateľa, ktorú topológiu sa snaží vymazať a jeho potvrdením sa daná topológia zo systému zmaže.



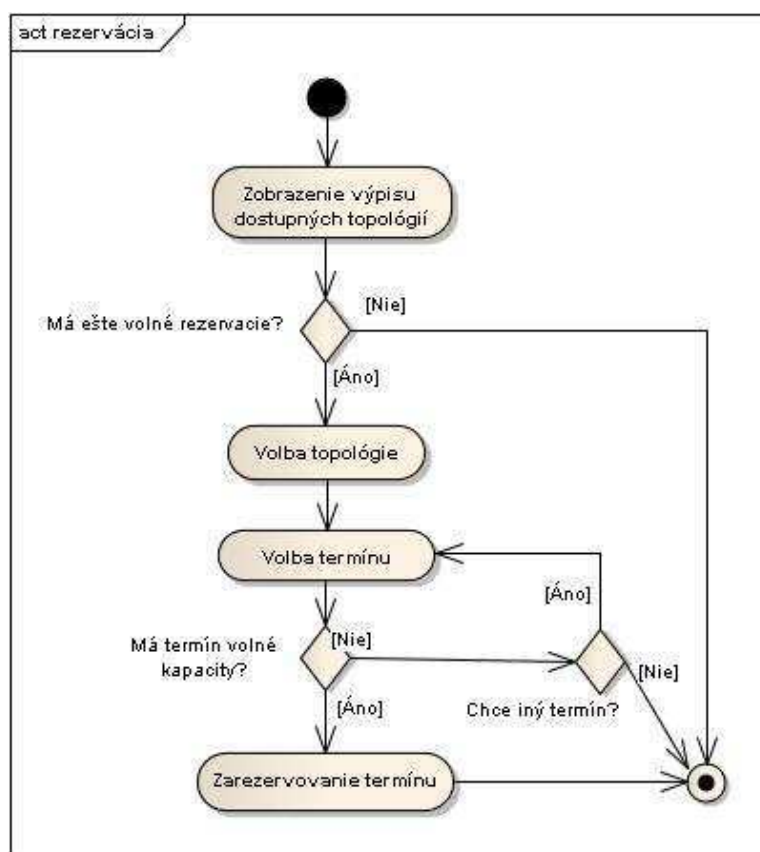
Obr. 2-4 Diagram aktivít správy topológií

#### 2.3.4 Rezervovanie a správa rezervácií topológií

Rezervačný systém funguje na princípe objednávania časových blokov pevnej dĺžky, ktorú môže z administratívneho menu nastaviť administrátor. Zmena dĺžky blokov je vykonávaná vždy na nasledujúci týždeň a mala by sa odvíjať od zložitosti topológií preberaných daný týždeň. Študent má možnosť si objednávať ním zvolené topológie v rámci daného týždňa, s obmedzením niekoľko hodín týždenne. Táto hodnota je takisto určovaná administrátorom.

Celý proces rezervácie je pomerne jednoduchý. Študent si zobrazí zoznam všetkých jemu dostupných topológií, vyberie si a následne mu je ponúknutý formulár s termínmi, ktoré majú ešte voľné kapacity, aby mohla byť uskutočnená rezervácia vybranej

topológie. Po zvolení termínu a potvrdení formulára je rezervácia zapísaná do databázy. Od tohto okamihu má študent okrem prístupných topológií na objednanie prístupný aj zoznam ním vykonaných rezervácií. Pomocou neho môže ktorúkoľvek zo svojich rezervácií zrušiť predtým, než nastane čas rezervovaného termínu. Celé zrušenie rezervácie prebieha jednoduchým formulárom, potvrdením ktorého sa zvolená rezervácia zmaže z databázy.



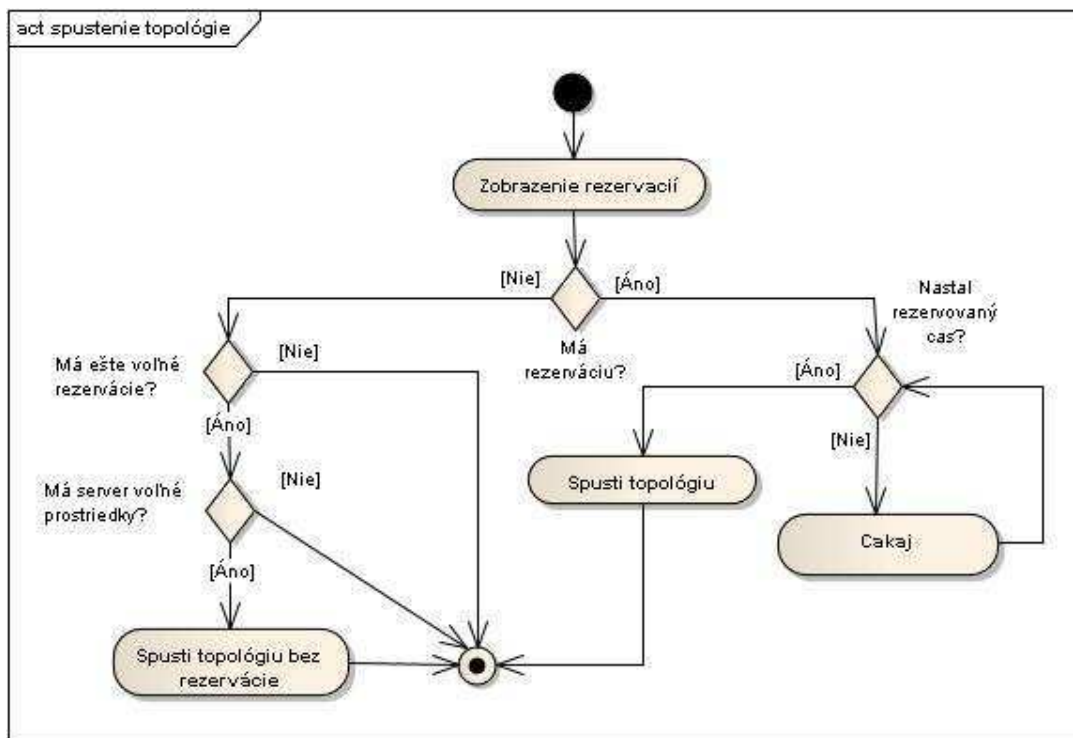
Obr. 2-5 Diagram aktivít pre rezerváciu

### 2.3.5 Spúšťanie a práca s topológiou

Táto funkcionality bude prístupná dvomi spôsobmi. Prvý je, že študent si zarezuje vybranú topológiu na určitý termín identifikovaný dátumom a časom začiatku a pri nastaní času tohto

termínu sa mu v zozname objednaných topológií pri konkrétnej položke objaví možnosť spustiť danú topológiu.

Druhým spôsobom je možnosť spustiť topológiu aj bez predošlej rezervácie, ale iba za podmienky, že užívateľ má ešte voľný počet hodín z hodín jemu pridelených na rezerváciu na daný týždeň a zároveň má server ešte voľné kapacity.

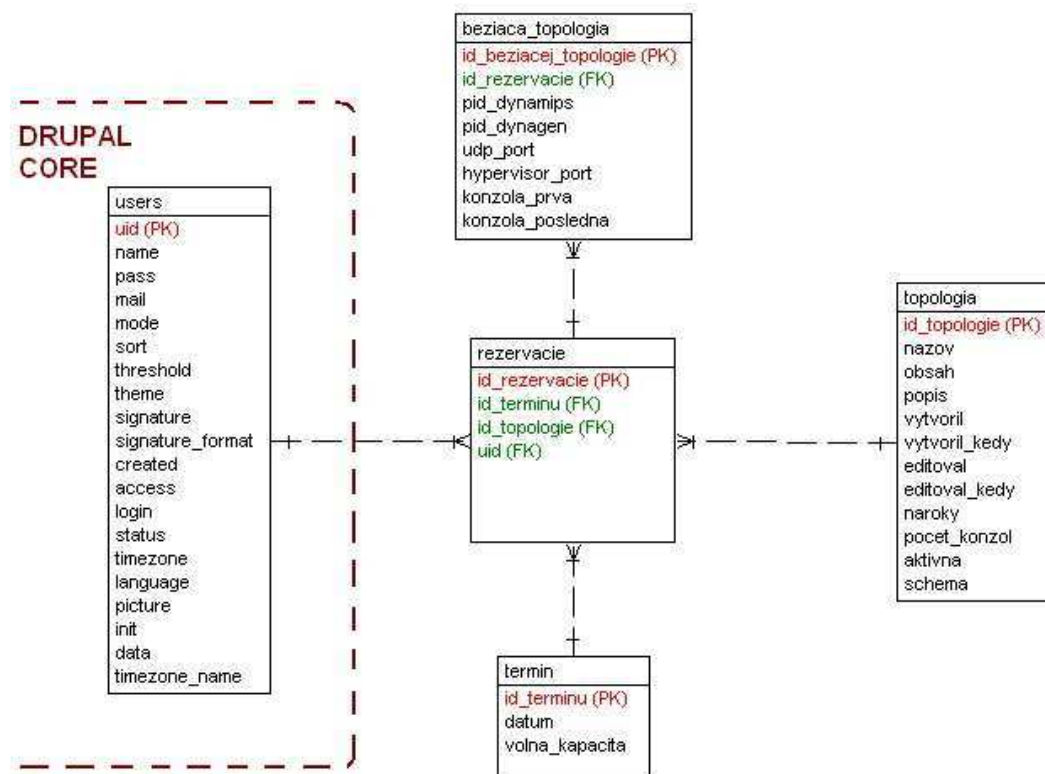


Obr. 2-6 Diagram aktivít pre spustenie topológie

Pri spustení topológie, systém zabezpečí spustenie danej topológie na serveri, informuje užívateľa o tom či všetko prebehlo v poriadku a sprístupní mu podstránku, odkiaľ sa môže na jednotlivé zariadenia pripojiť a pracovať s nimi. Pre ľahšiu orientáciu v topológii bude mať prístupnú schému zapojenia zariadení a počítaadlo udávajúce koľko času zostáva, kým mu vyprší termín. Po tomto čase systém užívateľa odpojí. Predtým však užívateľ môže svoju prácu uložiť, aby v nej mohol pokračovať neskôr.

## 2.4 Databáza

Na nasledovnom obrázku je možné vidieť navrhovanú štruktúru databázy.



Obr. 2-7 Databázový model

Databáza aplikácie je tvorená jadrom CMS Drupal a rozšírením v podobe dodatočných tabuliek. Databázové jadro Drupalu (na obrázku znázornené ako Drupal Core) je tvorené databázovými tabuľkami, ktorých hlavnou úlohou je uchovávanie všetkých potrebných údajov vyžadovaných CMS systémom Drupal. Bude slúžiť na zabezpečenie uchovávaní údajov o užívateľoch, užívateľských skupinách, oprávnení pre jednotlivé užívateľské skupiny a ďalších údajov potrebných pre chod rozhrania. Z dôvodu rozsiahlosti tu nebude uvedený popis jednotlivých databázových tabuliek jadra a ani jeho Dátový model.

Ako už bolo spomenuté pre zabezpečenie funkcionalít vyžadovaných touto aplikáciou, databázové jadro Drupalu nepostačuje a preto je potrebné ho rozšíriť o nasledovné tabuľky:

- *beziaca\_topologia*,
- *termin*,
- *topologia*,
- *rezervacie*.

Doplnené tabuľky budú napojené na zvyšnú časť databázy pomocou prepojenia tabuliek *users* a *rezervacie*.

### 2.4.1 Popis tabuliek

Tu sa nachádza podrobnejší popis databázových tabuliek a ich atribútov o ktoré bola databáza rozšírená.

#### **Termin**

V tejto tabuľke sú uchovávané jednotlivé termíny za celý týždeň, ktoré sú využívané na rezervovanie topológií.

- *id\_terminu* : je primárnym kľúčom tejto tabuľky, udržiava v sebe unikátny identifikátor daného termínu,
- *datum* : slúži na presnú identifikáciu začiatku daného časového bloku v rámci týždňa,
- *volna\_kapacita* : udržiava v sebe prístupnú kapacitu v MB, ktorú možno využiť na objednanie topológie pre daný časový blok.

#### **Beziaca\_konfiguracia**

V tejto tabuľke sa uchovávajú záznamy o aktuálne bežiacich topológiach, ktoré si užívatelia spustili.

- *id\_beziacej\_topologie* : je primárnym kľúčom tejto tabuľky, udržiava v sebe unikátny identifikátor bežiacej topológie,



- `id_rezervacie` : cudzí kľúč prenášaný z tabuľky rezervácie, odkazuje sa na rezerváciu, ktorá predchádzala danej spustenej topológii,
- `pid_dynamips` : hodnota ID procesu pridelená danej spustenej inštancii dynamipsu v rámci operačného systému,
- `pid_dynagen` : hodnota ID procesu pridelená danej spustenej inštancii dynagenu v rámci operačného systému,
- `udp_port` : základný port pre UDP NIO,
- `hypervisor_port` : číslo portu hypervisoru na ktorom bola spustená inštancia dynamipsu,
- `konzola_prva` : hodnota prvého prideleného portu z rozsahu pridelených portov pre konzoly použité v konfiguračnom súbore topológie,
- `konzola_posledna` : hodnota posledného prideleného portu z rozsahu pridelených portov pre konzoly použité v konfiguračnom súbore topológie.

## Topologia

V tejto tabuľke sú uchovávané všetky informácie o topológiach, ktoré sa v systéme nachádzajú.

- `Id_topologie` : je primárnym kľúčom tejto tabuľky, udržiava v sebe unikátny identifikátor danej topológie,
- `nazov` : názov, ktorým je daná topológia v systéme identifikovaná a je unikátny,
- `obsah` : text konfigurácie topológie, z ktorej bude vytvorený konfiguračný súbor použitý pri spúšťaní topológie,
- `popis` : stručný popis, ktorý má užívateľa informovať o čom daná topológia je,
- `vytvoril` : uchováva UID užívateľa, ktorý topológiu vytvoril,
- `vytvoril_kedy` : uchováva dátum a čas vytvorenia topológie,
- `editoval` : uchováva UID užívateľa, ktorý topológiu naposledy editoval,

- `editoval_kedy` : uchováva dátum a čas kedy bola topológia naposledy editovaná,
- `naroky` : počet MB, ktoré vyžaduje daná topológia zo systémových prostriedkov na svoje spustenie,
- `pocet_konzol` : počet konzolových portov vyžadovaných danou topológiou, zároveň určuje počet zariadení v topológii,
- `aktivna` : parameter určujúci či je daná topológia prístupná na spustenie alebo rezerváciu,
- `schema` : názov súboru s obrázkom použitým pre danú topológiu

### **Rezervacie**

V tejto tabuľke sú uchovávané všetky rezervácie, vykonané v rámci týždňa užívateľmi z dostupných topológií a termínov.

- `id_rezervacie` : je primárnym kľúčom tejto tabuľky, udržiava v sebe unikátny identifikátor danej rezervácie,
- `id_topologie` : cudzí kľúč odkazujúci sa na tabuľku topológie, informuje ktorú topológiu si užívateľ rezervoval,
- `id_terminu` : cudzí kľúč odkazujúci sa na tabuľku termín, informuje, ktorý termín si užívateľ rezervoval,
- `uid` : cudzí kľúč odkazujúci sa na tabuľku users, informuje o užívateľovi, ktorý vykonal danú rezerváciu.

### 3 Drupal

Drupal je PHP CMS systém vyvinutý Dries Buytaertom v roku 2000 pôvodne ako blogovací nástroj. Následným vývojom a sprístupnením projektu verejnosti na testovanie a pridávanie funkcionality sa Drupal rozvinul do dnešnej podoby. Napriek tomu, že je označovaný ako CMS systém, možno ho skôr označiť ako framework, kdeže na rozdiel od klasických CMS systémov sa zameriava hlavne na konfigurovateľnosť a prispôsobenie funkcionality. Zároveň nevyužíva klasický architektonický vzor Model-View-Controller (MVC), ale presentation-abstraction-control.

Základom Drupalu je stabilné a kvalitné jadro. To poskytuje základnú funkcionality, ktorá je pomocou API (Application Programming Interface) používaná ďalšími časťami systému. Základný stavebný prvok tvorí modul, keďže vo svojej podstate je Drupal modulárny systém. Svoju celkovú funkcionality získava z jadra a z pripojených modulov.

Moduly možno rozdeliť na:

- Základné moduly – moduly vytvárané pri inštalácii, ich počet závisí od konkrétnej verzie Drupalu. Tieto moduly sa delia na:
  - Core-required – moduly tvoriace jadro, nemožno ich zakázať. Jedná sa o moduly:
    - Block
    - Filter
    - Node
    - Sytem
    - User
  - Core-optional – zvyšné moduly, ktoré možno aktivovať aj deaktivovať
- Rozširujúce moduly – jedná sa o moduly, ktoré sú voľne dostupné k stiahnutiu (k dispozícii viac ako 4000), alebo vlastné užívateľom napísané moduly

Modulárny prístup poskytuje veľké možnosti ako si stránku jednoducho vyskladať z množstva jednotlivých modulov tak, aby poskytovala takmer akúkoľvek požadovanú funkcionality.

## 4 Implementácia

Ako už bolo spomenuté v kapitole návrh, pre zabezpečenie potrebnej funkcionality bolo potrebné vytvoriť vlastný modul, ktorý by slúžil ako rozšírenie webového rozhrania vytvoreného pomocou CMS Drupal, zabezpečujúce komunikáciu s programami Dynamips a Dynagen.

### 4.1 Úvod do problematiky

Napriek tomu, že je vytvorenie nového modulu v Drupale pomerne jednoduchá záležitosť, je potrebné dobre pochopiť spôsob tvorby modulu pre jeho správnu činnosť, tak aby neporušoval princípy na ktorých Drupal stavia. Drupal poskytuje vývojárom pomerne mohutné API, ktorým sprístupňuje svoje funkcie.

Celý modulový systém je postavený na tzv. *hookoch*. Hook je vlastne PHP funkcia v tvare *nazov\_funkcia()*, kde *nazov* predstavuje názov modulu a *funkcia* názov hooku. Každý takýto hook má sadu svojich parametrov a návratovú hodnotu. Zoznam a podrobný popis jednotlivých funkcií je možné nájsť na stránke Drupalu, konkrétne v sekcii venovanej API [6].

### 4.2 Základný koncept

Celá funkcionality modulu je obsiahnutá v bloku. Dôvodom pre toto riešenie bolo najmä dosiahnutie nezávislosti modulu od stránky. Pod nezávislosťou môžeme rozumieť, sprístupnenie celej poskytovanej funkcionality modulu jednoduchým povolením modulu a pridaním bloku vytvoreného našim modulom kdekoľvek na stránku.

## 4.3 Vytvorenie modulu

V tejto časti práce bude podrobnejšie popísaný postup tvorby modulu aj s uvedenými ukážkami kódu.

### 4.3.1 Vytvorenie súborov modulu

Vytvorenie modulu prebieha vytvorením dvoch potrebných súborov. Konkrétne sa jedná o súbory s príponami `.INFO` a `.MODULE`. Súbor `.INFO` obsahuje metainformácie popisujúce modul. Základný predpis tohto súboru v prípade modulu vytváraného v tejto práci je:

```
; $Id$  
name = Dynamips  
description = Modul na spustanie instancii dynamipsu.  
package = Networking  
core = 6.x
```

Môže obsahovať nasledovné polia:

- **Name** : je to povinná položka, ktorá udáva názov vytváraného modulu. Je potrebné, dodržať štandard Drupalu, čiže iba prvé písmeno prvého slova môže byť veľké.
- **Description** : povinná položka, obsahujúca krátky popis, ktorý sa zobrazuje v administračnej časti pri povoľovaní modulov. Môže mať maximálne 255 znakov.
- **Core** : povinná položka, vyjadruje verziu Drupalu pre ktorú je daný modul určený.
- **Package** : nepovinný, vyjadruje názov skupiny modulov do ktorej sa v rámci modulov zaradí.
- **Dependencies** : nepovinný, vyjadruje závislosť na inom module a Drupal nepovolí aktiváciu daného modulu, pokiaľ nie je aktivovaný aj modul uvedený v tejto položke.

Druhým potrebným súborom je súbor `.MODULE`. Tento súbor obsahuje implementáciu všetkých funkcií, ktoré by mali zabezpečovať požadovanú funkčnosť. Jedná sa prakticky o klasický php súbor, len s rozdielom, že neobsahuje uzatváraciu php značku a má príponu `.MODULE`.

### 4.3.2 Vytvorenie oprávnení

Prvou vecou, ktorú bolo potrebné v module implementovať boli oprávnenia, pomocou ktorých by sa pridelovala funkčnosť jednotlivým používateľským skupinám, aby sa zabránilo prístupu k funkciám, ktoré by daná používateľská skupina nemala mať prístupné. Na definovanie skupiny oprávnení bola použitá funkcia `dynamips_perm()`. Návratovou hodnotou tejto funkcie je pole reťazcov, obsahujúce všetky potrebné typy oprávnení pre náš modul. Funkcia zabezpečuje sprístupnenie vytvorených oprávnení na stránke s oprávneniami, odkiaľ ich je možné používať a priradovať práva jednotlivým užívateľským skupinám.

### 4.3.3 Vytvorenie položiek menu

Ďalšou potrebnou časťou modulu je definovanie položiek menu, aby boli jednotlivé funkcie prístupné pomocou URL adresy (Uniform Resource Locator). Na to slúži funkcia `dynamips_menu()`. Opäť sa jedná o pole, v ktorom sú definované jednotlivé položky menu. Každá položka má v sebe definovanú cestu, názov, typ, oprávnenia na prístup a popis. Názorná ukážka časti menu zobrazuje položku menu, ktorá zabezpečuje zobrazenie formulára na vytváranie novej topológie:

```
function dynamips_menu() {  
    $items = array();  
    $items['dynamips/new_conf'] = array(  

```

```

    'title' => 'Create new configuration file',
    'page callback' => 'dynamips_new_conf',
    'access arguments' => array('tvorba konfiguracie'),
    'description' => t('Create new configuration file'),
    'type' => MENU_CALLBACK,
  );
  return $items;
}

```

Takýmto spôsobom je cesta `dynamips/new_conf` zaregistrovaná. Pre funkčnosť danej cesty je však potrebné definovať funkciu volanú prehliadačom, ktorá vygeneruje konkrétny obsah. Keďže sa má jednať o obsah vo forme formuláru, docielime jeho vytvorenie volaním funkcie:

```

function dynamips_new_conf() {
  return drupal_get_form('dynamips_new_conf_form');
}

```

Funkcia `drupal_get_form($form_id)` sa postará o zavolanie konštruktora formulára, alebo získanie formulára z cache, ak už formulár bol vytvorený.

Následne je potrebné zadať tzv. form builder, čo je vlastne php funkcia, ktorej názov je rovnaký ako `$form_id` pri volaní `drupal_get_form($form_id)`. Návrátovou hodnotou form buildera je pole popisujúce štruktúru formulára, ktorý sa má vytvoriť.

Nakoniec je ešte potrebné definovať submit funkciu:

```

function dynamips_new_conf_form_submit($form, $form_state) {
  //kod submit funkcie
}

```

Pred samotným potvrdením je však dobré vykonávať validáciu. Tú v tomto konkrétnom prípade možno zabezpečiť funkciou:

```

function dynamips_new_conf_form_validate($form, &$amp;$form_state) {

```



```

    // kod validacie co sa ma vykonat
}

```

Ako je vidno obe funkcie majú rovnaký základ názvu ako aj funkcia form buildera, líšia sa iba pridaním hookov `_submit` a `_validate`. Týmto spôsobom Drupal vie, že dané funkcie sa viažu ku konkrétnemu formuláru.

Spôsobom, ktorý bol popísaný sú vytvorené aj zvyšné funkcie, ktoré webové rozhranie poskytuje.

#### 4.3.4 Definovanie bloku

Ako bolo na začiatku kapitoly spomenuté, celá funkcionálnosť je zabalená do bloku. Na jeho zdefinovanie je použitá funkcia:

```

function dynamips_block($op = 'list', $delta = 0, $edit = array()) {
  //kod bloku
}

```

Vyžaduje tri parametre:

- \$op: vyjadruje jednu zo štyroch operácií:
  - list – zoznam všetkých blokov definovaných modulom,
  - view – zobrazenie obsahu bloku,
  - save – uloženie nastavení,
  - configure – formulár s konfiguráciou bloku,
- \$delta : tento parameter slúži na identifikáciu bloku, keďže modul môže definovať viacej blokov, je potrebné ich odlíšiť,
- \$edit : používa sa na ukladanie dát v prípade, že \$op má hodnotu save.

Použitie tejto funkcie a definovaním všetkých jej položiek má za následok, že vo webovom rozhraní je prístupný blok, ktorý obsahuje administratívnu časť slúžiacu na nastavenie hlavných nastavení celého systému ako napr. dĺžka bloku prístupného na

rezerváciu, dĺžka obdobia na ktorú je možné vykonávať rezervácie,  
počiatočné porty zariadení a podobne.

## Záver

Webová aplikácia Virtual Lab ponúka možnosť študentom a učiteľom jednoducho a pohodlne pracovať s virtuálnymi topológiami. K dispozícii majú prehľadné rozhranie, ktoré učiteľom dáva možnosť vytvárať nové topológie, alebo editovať už existujúce a sprístupňovať ich študentom, ktorý by s nimi následne mohli pracovať a tým zdokonaľovať svoje praktické zručnosti z oblasti počítačových sietí. Aplikácia zároveň ponúka rezervačný systém, ktorý zabezpečí každému študentovi garantovaný počet hodín použiteľných na prácu so systémom. Systém je navrhnutý tak, aby sa podľa potrieb dali meniť jednotlivé jeho nastavenia ako dĺžka blokov, počet garantovaných hodín a podobne. Napriek tomu, že aplikácia ponúka požadovanú funkcionálnosť, je plne rozšíriteľná o nové funkcie podľa potrieb užívateľov.

## **Zoznam použitých zdrojov**

- [1] [http://en.wikipedia.org/wiki/Packet\\_Tracer](http://en.wikipedia.org/wiki/Packet_Tracer)
- [2] <http://www.boson.com>
- [3] HAMŠÍK, A., SCHMOTZER, M., MÓZES, T., PÉTI, P., NEMSILA, M.: Simulátor komunikácie v počítačovej sieti, November 2007
- [4] SOUSEDEK, J.: Simulátor sítě ve výuce, August 2007
- [5] VESELÝ, V.: Dynagen/Dynamips emulátor Cisco routeru
- [6] <http://api.drupal.org/api/>
- [7] [http://www.ipflow.utc.fr/index.php/Cisco\\_7200\\_Simulator](http://www.ipflow.utc.fr/index.php/Cisco_7200_Simulator)
- [8] <http://interval.cz/clanky/vyvoj-modulu-pro-cms-system-drupal-6-x/>
- [9] <http://dynagen.org/tutorial.htm>

## **Zoznam skratiek**

API	Application Programming Interface
CMS	Content Management System
CPU	Central Processing Unit
FR	Frame Relay
HDLC	High-Level Data Link Control
IOS	Internetwork Operating System
JIT	Just-In-Time
JTA	Java Telnet Application
MIPS	Microprocessor without Interlocked Pipeline Stages
MVC	Model-View-Controller
PPP	Point-to-Point Protocol
PT	Packet Tracer
RAM	Random Access Memory
URL	Uniform Resource Locator

## 5 Príloha

### 5.1 Správne písanie konfiguračného súboru

Pre správne fungovanie konfiguračných súborov písaných cez webové rozhranie systému, je potrebné dodržať nasledovné pravidlá:

- adresa servera musí mať tvar *[localhost: hypervisorPort]*,
- základný port pre UDP NIO musí mať tvar *udp = udpPort*,
- konzolové porty musia mať tvar *console = consPortX* , pričom X je číslo od 1 až po počet konzol vyskytujúcich sa v topológií.

Tieto premenné sa používajú z dôvodu dynamického vytvárania konfiguračného súboru na serveri, kedy sú programom nahradené reálnymi hodnotami portov.

Správny zápis konfiguračného súboru uvádzaného v kapitole 1.4.2 *Dynagen*, by mal vyzeráť nasledovne:

```
# Simple lab

ghostios = True
sparsemem = True

[localhost:hypervisorPort]

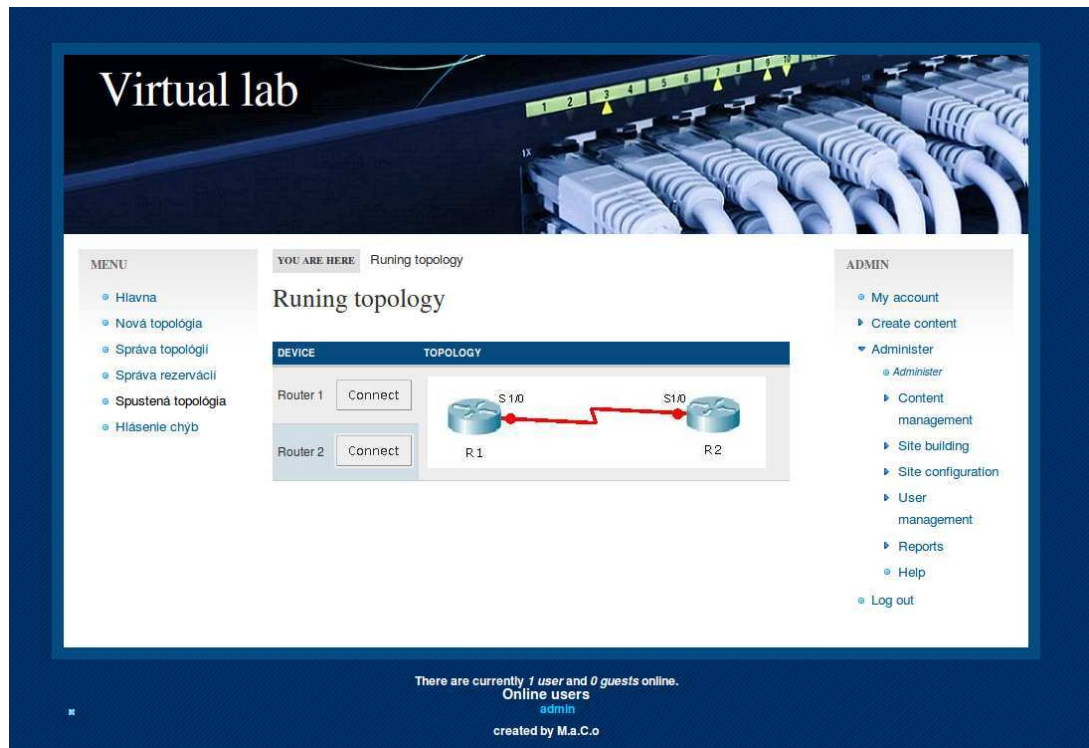
udp = udpPort

[[3620]]
    image = /opt/7200-images/c3620-i-mz.123-22.bin.unp

[[ROUTER R1]]
    s1/0 = R2 s1/0
    console = consPort1

[[ROUTER R2]]
    console = consPort2
```

## 5.2 Ukážka webového rozhrania



Obr. 5-1 Ukážka webovej aplikacie