

Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 812 19 Bratislava

# Simulátor komunikácie v počítačovej sieti

Tímový projekt

**Tím č. 9:** Bc. Bielik Martin  
Bc. Gula Štefan  
Bc. Marko Peter  
Bc. Michalec Vladimír  
Bc. Schusterová Zuzana

Vedúci projektu: Ing. Grellneth Igor, PhD.

Školský rok: 2008/2009

# Obsah

Obsah .....	ii
Zoznam obrázkov .....	v
1. Úvod .....	1
1.1. Účel a rozsah dokumentu .....	1
1.2. Prehľad dokumentu .....	1
1.3. Zadanie projektu.....	1
1.4. Slovník pojmov .....	2
1.5. Použité skratky .....	3
2. Analýza.....	5
2.1. PIX/ASA Firewall .....	5
2.1.1. Typy firewall-ov .....	5
2.1.2. PIX.....	8
2.1.3. ASA.....	9
2.1.4. PIX/ASA.....	9
2.2. Simulátor verzus emulátor .....	10
2.3. Boson Netsim .....	10
2.4. Packet Tracer .....	12
2.5. Dynamips.....	13
2.6. VLAB .....	14
2.6.1. Funkcionalita .....	14
2.6.2. Štartovacie skripty .....	16
2.6.3. Databáza .....	18
2.6.4. Proxy.....	20
2.6.5. Prepínač.....	20
2.6.6. Web rozhranie .....	20
3. Špecifikácia .....	22
3.1. Zmeny vo VLAB .....	22
3.1.1. Štartovacie skripty .....	22
3.1.2. Proxy.....	22

3.2. Pridané prvky .....	23
3.2.1. Manažovateľný prepínač.....	23
3.2.2. PIX Firewall .....	23
4. Návrh .....	24
4.1. Štartovacie skripty .....	24
4.2. Proxy server.....	25
4.3. Manažovateľný prepínač.....	26
4.4. PIX Firewall .....	26
4.5. Laboratórne cvičenia .....	26
5. Implementácia.....	28
5.1. Štartovacie skripty .....	28
5.2. Proxy server.....	30
5.2.1. SSH server .....	30
5.2.2. Telnet klientska aplikácia .....	31
5.2.3. Shell skript.....	31
5.2.4. Prepojenie celého systému proxy servera.....	32
5.3. Manažovateľný prepínač.....	33
5.4. PIX Firewall .....	34
5.5. Laboratórne cvičenia .....	35
6. Testovanie a overenie riešenia .....	40
6.1. Proxy server.....	40
6.1.1. Overenie bezpečnosti použitých aplikácií .....	40
6.1.2. Testovanie záťaže servera .....	40
6.1.3. Overenie bezpečnosti niektorých z aplikácií .....	40
6.2. Manažovateľný prepínač.....	41
6.3. PIX Firewall .....	41
6.4. Laboratórne cvičenia .....	42
7. Záver .....	43
8. Zoznam literatúry.....	44
9. Prílohy .....	45

**Príloha A - Riadenie projektu**

**Príloha B - Posudky**

**Príloha C - Používateľská príručka**

**Príloha D - Inštalačná príručka**

**Príloha E - Článok do časopisu**

**Príloha F - Obsah priloženého CD**

## Zoznam obrázkov

Obrázok 2.1: Ukážka prostredia programu Boson NetSim. ....	12
Obrázok 2.2: Ukážka prostredia programu Packet Tracer. ....	13
Obrázok 2.3: Ukážka konzoly programu Dynamips. ....	14
Obrázok 2.4: Fyzický model údajov [1]. ....	19
Obrázok 4.1: Prepojenie <i>proxy</i> servera s ostatnými komponentmi. ....	25
Obrázok 5.2: Prepojenie <i>proxy</i> servera s ostatnými komponentmi. ....	32
Obrázok 5.3: Úrovně laboratórných cvičení. ....	35
Obrázok 5.4: Zoznam laboratórných cvičení. ....	37
Obrázok 5.5: Topológia cvičenia „Basic BGP“ ....	38

# 1. Úvod

## 1.1. Účel a rozsah dokumentu

Dokument obsahuje analýzu, špecifikáciu a návrh softvérového systému určeného na simuláciu komunikácie v počítačovej sieti. Je výsledkom priebežnej práce na tímovom projekte, ktorý je súčasťou predmetu „Tímový projekt“ v študijnom odbore Počítačové systémy a siete na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave. Tento dokument je určený predovšetkým pre študentov a pedagógov spomínanej fakulty.

## 1.2. Prehľad dokumentu

Dokument je rozčlenený do jednotlivých kapitol podľa potreby. V prvej kapitole je uvedený účel a rozsah dokumentu, úplné znenie zadania projektu a význam skratiek, ktoré dokument obsahuje. V druhej kapitole je opísaná podrobná analýza existujúcich riešení s danou problematikou. Analyzuje hlavne riešenie, ktoré vytvorili študenti fakulty na tomto predmete minulý rok. Tretia kapitola obsahuje špecifikáciu požiadaviek. Štvrtá kapitola rozoberá návrh riešenia požiadaviek, ktoré boli uvedené v predchádzajúcej kapitole. Posledná kapitola uvádza všetky zdroje literatúry, ktoré prispeli k vytvoreniu tejto dokumentácie.

## 1.3. Zadanie projektu

Navrhните a zrealizujte programový systém pre simuláciu sieťovej komunikácie na druhej a tretej vrstve sieťovej architektúry RM OSI.

System má umožňovať:

- definovanie topológie simulovanej siete,
- simuláciu rôznych prepájacích zariadení (napr. prepínač, smerovač, *firewall*),
- simuláciu komunikácie medzi prepájacími zariadeniami.

Funkčnost navrhnutého systému overte v sieti so simulovanými zariadeniami pomocou komunikácie medzi koncovými zariadeniami.

## 1.4. Slovník pojmov

Tu je vypísaný zoznam pojmov, ktoré používame v dokumente a nemajú v slovenskom jazyku ekvivalentný preklad. Skloňovanie týchto výrazov v texte je spravené cez pomlčky, čiže základ slova ostáva vždy nezmenený. Všetky výrazy sú uvádzané v texte *kurzívou*.

<i>Applet</i>	je aplikácia, ktorá sa spúšťa z iného programu napríklad z webového prehliadača.
<i>Bash</i>	je unixový príkazový shell interpret naprogramovaný v rámci projektu GNU. Názov je akronym k názvu <b>B</b> ourne <b>a</b> gain <b>s</b> hell - je založený na Bourne Shellu (bsh), čo bol najpoužívanejší unixový shell
<i>Demo</i>	je predvádzací program.
<i>Enterprise</i>	v preklade podnik, podniková.
<i>Ethernet</i>	je technológia počítačových sietí.
<i>Firewall</i>	je sieťové zariadenie a/alebo softvér, ktorého úlohou je oddeliť siete s rôznymi prístupovými právami a kontrolovať tok dát medzi týmito sieťami.
<i>Flash pamäť</i>	celým menom Flash-EEPROM, je zrýchlená elektricky preprogramovateľná pamäť ROM (Read-Only Memory).
<i>Frame Relay</i>	jeho podstata spočíva v zriadení prístupových okruhov a v zriadení virtuálnych okruhov.
<i>Hypervisor</i>	je konzola programu Dynamips.
<i>Proxy</i>	je špeciálny typ servera - prostredníka v komunikácii, ktorý sa umiestňuje medzi klienta a servery, s ktorými komunikuje. <i>Proxy server</i> sa tvári voči klientovi ako server a voči serveru ako klient.

<i>Root</i>	alebo inak administrátor má schopnosť pristupovať ku všetkým súborom v systéme.
<i>Telnet</i>	je sieťový komunikačný protokol, ktorý umožňuje pripojenie k inému zariadeniu na sieti.
<i>Wrapper</i>	je mechanizmus, ktorý umožňuje riadiť prístup k službám vášho servera na základe adresy, z ktorej prichádza požiadavka klienta.

## 1.5. Použité skratky

V tejto časti uvádzame abecede zoradené všetky použité skratky, aby si nikto nezamenil niektoré zo skratiek.

ACL	Access Control List
ASA	Adaptive Security Appliance
CCNA	Cisco Certified Network Associate
CCNP	Cisco Certified Network Professional
ID	Identification Database
IOS	Internetwork Operating System
IP	Internet Protocol
IPS	Intrusion Prevention Sensor
IPv6	Internet Protocol version 6
ISDN	Integrated Services Digital Network
LAN	Local Area Network
NAT	Network Address Translation
OS	Operating System
OSPF	Open Shortest Path First
PAT	Port Address Translation



RFC	Request for Comments
RIP	Routing Information Protocol
RM OSI	Reference Model Open System Interconnected
SPI	Stateful Packet Inspection
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UML	Unified Modeling Language
VDE	Virtual Distributed Ethernet
VLAB	Virtual Lab
VLAN	Virtual LAN
VPN	Virtual Private Network
WAN	Wide Area Network

## 2. Analýza

V tejto časti sa budeme zaoberať predovšetkým analýzou existujúcich riešení daného problému, špeciálne riešenie študentov tímu č. 1 z roku 2007/2008 s názvom VLAB - Virtuálne laboratórium. Zameriame sa na funkcionality programu, štartovacie skripty, databázu, riešenie *proxy* servera a samotného web rozhrania.

Analýza problematiky počítačových sietí ako napríklad:

- druhy sietí (LAN, WAN, atď.),
- RM OSI a jeho vrstvy,
- TCP/IP model,
- komunikácia v sieti,
- sieťové prvky (prepínače, smerovače, atď.),

je rozobratá v dokumentácií k projektu z minulého roku a je dostupná na stránke projektu [1]. Preto sa radšej venujeme analýze PIX/ASA *firewall-u*, ktorý by sme chceli neskôr implementovať do systému.

### 2.1. PIX/ASA Firewall

#### 2.1.1. Typy firewall-ov

Všetky *firewall-y* môžeme rozdeliť do týchto troch kategórií [3]:

- paketové filtre,
- *proxy* filtre,
- stavové paketové filtre.

### **Paketové filtre**

*Firewall*, ktorý spadá do kategórie filtrov TCP/IP, spravidla analyzuje sieťovú komunikáciu na transportnej vrstve, prípadne na sieťovej (internetovej) vrstve sady protokolov TCP/IP. To znamená, že môže v sieti filtrovať ľubovoľné prenášané dáta, ktoré zodpovedajú štandardnej sade protokolov TCP/IP (prípadne inej štandardnej rodine protokolov). Toto je možné len preto, lebo jednotlivé polia každého dátového paketu sú známe z ich štandardov, ktoré nájdeme v RFC [2]. Paketový filter teda analyzuje statické informácie, ktoré obsahuje hlavička paketu.

Pri konfigurácii paketového filtra vytvárame pravidlá, ktoré sú definované určitými kritériami pre zdroj, cieľ alebo zdroj aj cieľ komunikácie. Tento filter môže povoľovať alebo zamietajú posielanie dát na základe jedného alebo viacerých kritérií:

- zdrojová IP adresa,
- cieľová IP adresa,
- protokol,
- zdrojový port,
- cieľový port.

Príkladom paketového filtra môže byť bežný Cisco smerovač, ktorý má v konfigurácii zapnuté filtrovanie dátovej komunikácie s definovaným prístupovým zoznamom (ACL - access control list).

Dôležitou vlastnosťou takéhoto bežného filtra je, že si neudržiava žiadne stavové informácie. To znamená, že rozhodne podľa platných prístupových zoznamov o povolení, alebo naopak zamietnutí prijatého paketu, ale po jeho spracovaní si k nemu nezaznamená žiadne informácie. Pri nasledovnom prijatom pakete sa celý proces opakuje ako keby od začiatku.

Tento druh filtra má určité nevýhody:

- Filtrom môžu prechádzať ľubovoľne zostavené pakety, ktoré iba zodpovedajú kritériám prístupových zoznamov.
- Pakety môžu prejsť filtrom taktiež vo fragmentovanej podobe.
- Správne vytvoriť, implementovať a udržiavať zložité prístupové zoznamy býva náročné.

- Niektoré služby sa nedajú filtrovať, keď nevieme vopred číslo portu, ktoré služba používa.

### Proxy filtre

*Proxy* filter je *firewall-ové* zariadenie, ktoré skúma pakety z pohľadu vyšších vrstiev referenčného modelu OSI. Obvykle je to na štvrtej až siedmej vrstve. Vďaka tomu je *proxy* filter veľmi silný, ale zároveň tým znižuje priepustnosť siete pri komunikácii medzi koncovými zariadeniami. Toto zariadenie skrýva cenné dáta pred vonkajším svetom, pretože používatelia musia komunikovať so zabezpečeným systémom iba prostredníctvom *proxy* serveru. Používateľ musí pre získanie prístupu do siete prejsť špeciálnym procesom, v ktorom sa ustanoví stav relácie, prebehne autentizácia používateľa a uplatnia sa prístupové práva. To znamená, že používatelia vnútornej siete sa ku vonkajším službám pripájajú cez určité aplikačné programy, ktoré bežia na bráne a prepájajú vnútornú sieť s vonkajšou nechránenou zónou a sprostredkujú tak ich potrebnú komunikáciu.

Pri jednom zo spôsobov činnosti *firewall-u* s *proxy* filtrom musí používateľ vo vnútornej sieti najprv vytvoriť reláciu so samotným *firewall-om*. Potom sa používateľ musí autentizovať a podľa jeho ID a hesla mu systém povolí okruh prístupových práv do vonkajšej siete. V tomto modeli sa vždy vytvoria dve samostatné relácie medzi používateľom a *proxy* serverom, a druhá medzi *proxy* serverom a požadovaným cieľom.

*Proxy firewall* môže fungovať ešte ďalším spôsobom. Používateľ vnútornej siete pri ňom vytvára jedinou reláciu do cieľa umiestneného vo vonkajšej sieti. Takto sa to aspoň javí samotnému používateľovi. V skutočnosti *proxy firewall* do takto nadviazanej relácie vstúpi, podľa určitých informácií (napríklad zdrojová IP adresa), vykoná autentizáciu a nakoniec opäť vytvorí dve relácie. Z pohľadu používateľa je tento spôsob oveľa transparentnejší.

S *proxy firewall-om* sú spojené nasledovné problémy:

- *Proxy firewall* tvorí v sieti jediné zraniteľné miesto. Ak niekto napadne prístup k samotnému *firewall-u*, je ohrozená bezpečnosť celej siete.
- Do *firewall-u* sa ťažko dopĺňajú nové služby.
- Pod zaťažením pracuje pomalšie.

### Stavové paketové filtre

Tretí typ *firewall-u* spojuje tie najlepšie vlastnosti z predchádzajúcich dvoch filtrov. Stavový paketový filter si pre každú reláciu, ktorá vedie cez *firewall*, pamätá kompletne stavové informácie. Pri každom nadviazaní nového spojenia, či už v prichádzajúcom alebo odchádzajúcom smere, sa do stavovej tabuľky relačného toku zaznamenajú príslušné údaje. Túto metódu filtrovania používa aj Cisco PIX Firewall.

Stavová tabuľka relačných tokov obsahuje ku každému spojeniu TCP/UDP informácie o zdrojovej a cieľovej adrese, čísla portov, údaje o poradových číslach TCP a prípadné doplňujúce príznaky. S nadviazaním relácie cez *firewall* sa vytvorí spojenie a všetky prichádzajúce i odchádzajúce pakety sa následne porovnávajú s relačným tokom, zaznamenaným v stavovej tabuľke relačných tokov, a priechod dát sa povolí len v prípade, že k nim existuje príslušné spojenie.

Táto metóda pri filtrovaní je veľmi účinná z nasledovných dôvodov:

- Pracuje nad jednotlivými paketmi a každý z nich porovnáva voči nadviazaným komunikačným spojeniam.
- Je rýchlejší ako filtrovanie paketov alebo *proxy server*.
- Ku každej transakcii (UDP alebo TCP) si *firewall* zaznamenáva určité údaje do tabuľky, ku ktorej sa môže neskôr vrátiť a môže tak stanoviť, či prijatý paket prislúcha k niektorému spojeniu, alebo pochádza z neoprávneného zdroja.

### 2.1.2. PIX

PIX je hardvérový *firewall* vyrábaný spoločnosťou Cisco. Jeho hlavnou úlohou je filtrovať komunikáciu medzi viacerými fyzickými rozhraniami (alebo aj logickými rozhraniami pri použití VLAN). Funguje na princípe stavového *firewall-u* (SPI) - keď dorazí paket na rozhranie s vyššou bezpečnostnou úrovňou, vytvorí sa záznam do stavovej tabuľky na základe protokolu štvrtej vrstvy OSI a príslušných portov. Paket potom pošle na rozhranie nižšej bezpečnostnej úrovne. Keď bude druhá strana posilať odpoveď na takýto paket, na základe stavovej tabuľky budú tieto odpovede preposlané z menej bezpečného rozhrania na to bezpečnejšie. Ak by sa v stavovej tabuľke záznam nenašiel, pakety budú zahodené (pokiaľ ich

administrátor implicitne nepovolí). Okrem tejto funkcie vykonáva PIX aj preklad adres (NAT) alebo portov (PAT). Je možné použiť ho ako počiatočný alebo koncový bod VPN spojenia. Dokáže tiež kontrolovať obsah protokolu v aplikačnej vrstve paketu (je tak možné zabrániť niektorým typom útokov). Samozrejme je možné filtrovať komunikáciu podľa veľkého množstva parametrov.

### 2.1.3. ASA

Nástupca PIX Firewall-u prišiel v podobe Cisco ASA (Adaptive Security Appliance). Toto zariadenie preberá všetky doterajšie funkcie PIX-u, obsahuje však modernejší a rýchlejší hardvér a nový operačný systém založený na jadre Linuxu. Navyše ponúka možnosť použitia zásuvných modulov, ktoré pridávajú funkcie IPS zariadenia (Intrusion Prevention Sensor) alebo antivírového skenera. Taktiež priniesla ASA možnosť využiť SSL WebVPN pripojenia, tieto umožňujú pripájať sa pomocou technológie VPN na zariadenie bez použitia natívneho klienta, postačuje prehliadač webových stránok.

### 2.1.4. PIX/ASA

PIX/ASA funguje ako samostatný hardware architektúry x86, na ktorom beží špeciálny operačný systém. Spravovať je ho možné, podobne ako iné zariadenia firmy Cisco, pomocou textovej konzoly, grafického rozhrania, alebo protokolov *Telnet* a SSH. Funguje čiastočne aj ako smerovač. Podporuje statické smerovanie a protokoly RIP a OSPF.

Vďaka hardvérovej architektúre týchto zariadení je možné emulovať ich činnosť pod rôznymi OS na klasických stolných počítačoch. Slúži na to program Qemu, ktorý emuluje rôzne druhy procesorov (a príslušných periférií) a je multiplatformový. Priamo pre emuláciu PIX Firewall-u existuje voľne dostupná modifikácia tohto programu s názvom Pemu. Spúšťa sa pomocou príkazového riadku s parametrami špecifikujúcimi virtuálne sieťové karty PIX-u:

- číslo portu, pod ktorým bude prístupná virtuálna konzola PIX-u,
- umiestnenie virtuálnej *flash* pamäte.

Takto spustená emulácia PIX Firewall-u sa môže pomocou tunelov prepojiť s emulátorom Dynamips (podrobnejší popis emulátoru nájdete v kapitole 2.5). Novšie verziu Dynagen-u už majú program Pemu zakomponovaný priamo v sebe, takže emulácia je ešte jednoduchšia (stačí jednoduchá modifikácia súboru s topológiou, veľmi podobná zápisu samotného Dynagen-u). Pre správny beh emulácie potom už stačí iba spustiť Pemu server.

Emulácia zariadenia ASA je tiež možná, avšak program Pemu ju nepodporuje, takže treba používať pôvodný program Qemu. Je teda treba zložitejšie nastavovanie parametrov behu programu a je potrebné nastaviť spojenie pomocou tunelov aj na strane Dynagen-u.

Funkcionalita zariadení PIX a ASA je však v súčasnej verzii ich OS prakticky rovnaká (do virtuálnych zariadení nemožno pridať zásuvné moduly, čo je hlavný rozdiel ASA oproti PIX-u). Takže v našom riešení sa budeme venovať len emulácii pomocou programu Pemu.

## 2.2. Simulátor verzus emulátor

Hlavný rozdiel medzi simulátorom a emulátorom je v tom, že simulátor sprístupňuje funkcionality implementovanú priamo v programovom kóde. Emulátor na druhej strane väčšinou vytvára len virtuálne prostredie, alebo hardvér na nízkej úrovni, ktorý potom umožňuje spustiť reálny systém. Príkladom emulátora počítača sú napríklad programy VmWare alebo Virtual PC.

## 2.3. Boson Netsim

Boson Software je spoločnosť, ktorá sa zaoberá podporou vo vzdelávaní a certifikovaní profesionálov v oblasti IT. Jedným z produktov, ktoré táto spoločnosť na trhu poskytuje sú práve sieťové simulátory. Tieto simulátory sú primárne určené pre záujemcov o CCNA alebo CCNP certifikáciu. Pomocou množstva pred pripravených laboratórnych úloh cielene pripravujú záujemcov na skúšky z týchto uznávaných priemyselných certifikátov. Z rozdielnej zložitosti týchto dvoch skúšok vyplývajú aj dve rozdielne verzie simulátorov. Z internetu je možné zadarmo stiahnuť *demo* verziu NetSim pre CCNA. Boson Netsim je aplikácia, ktorá umožňuje výstavbu rôznych sieťových topológií, obsahujúcich rôzne typy smerovačov,

prepínačov a pracovných staníc. Návrh topológie a konfigurovanie zariadení sú dva oddelené programy. Tento simulátor pozostáva z troch častí:

### **1. Boson Lab Navigator**

Obsahuje štruktúrovaný zoznam vytvorených konfigurácií, ktoré môže študent použiť na precvičenie rôznych funkcií. Sú tu návody stručne popisujúce jednotlivé príkazy od úplného začiatku. Ku každému návodu je vytvorené zapojenie zariadení, ktoré je možné rovno spustiť v simulátore a otestovať si nadobudnuté vedomosti.

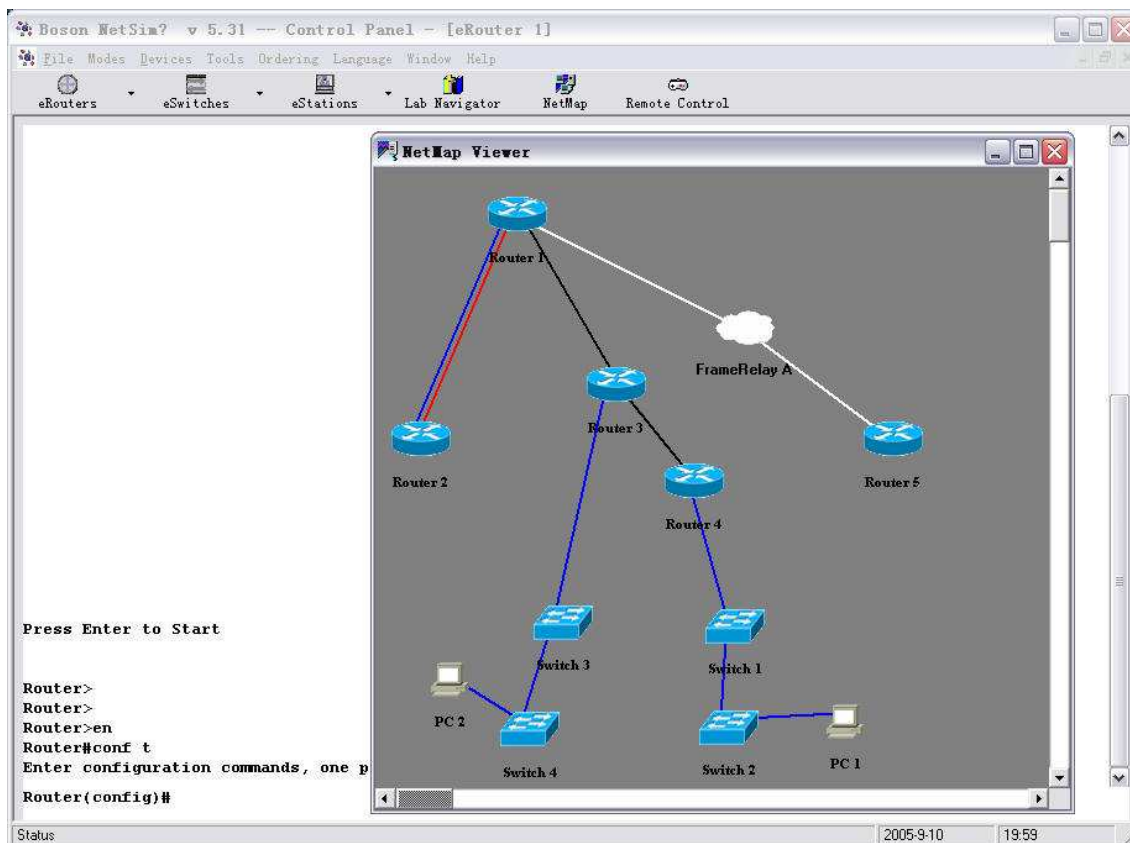
### **2. Boson Network Designer**

Sieť je budovaná „*drag&drop*“ technikou spojenou s jednoduchými sprievodcami pridaním daného zariadenia do topológie. Aplikácia podporuje Cisco smerovače sérií 800, 1000, 1600, 1700, 2500, 2600, 3600 a 4500, prepínače sérií 1900, 2900, 3500 a 5000. Tieto zariadenia je možné spojiť pomocou *Ethernet*, sériových a ISDN prepojení. Zaujímavejšou časťou aplikácie je časť umožňujúca konfigurovať sieťové zariadenia. Konfigurácia zariadení je realizovaná pomocou konzolových príkazov známych z Cisco IOS. Aplikácia má jednoduché grafické používateľské rozhranie, čo spolu s jednoduchými sprievodcami pre pridanie zariadení používateľ ocení, najmä pri tvorbe rozľahlejšej topológie.

### **3. Boson NetSim**

Vykonáva samotnú simuláciu topológie. Po načítaní zapojenia ponúka zoznam zariadení a možnosť ich konfigurovať. Pripojenie na konzolu je možné realizovať v okne tejto aplikácie, alebo prístupom cez *Telnet* v samostatnom okne. Do hlavného okna simulátora sa môže načítať vytvorená topológia. Jednotlivé prvky siete môžeme konfigurovať v grafickom prostredí programu pomocou konzoly. Veľkou výhodou simulátora je možnosť riešiť niektorú z prednastavených sieťových topológií.

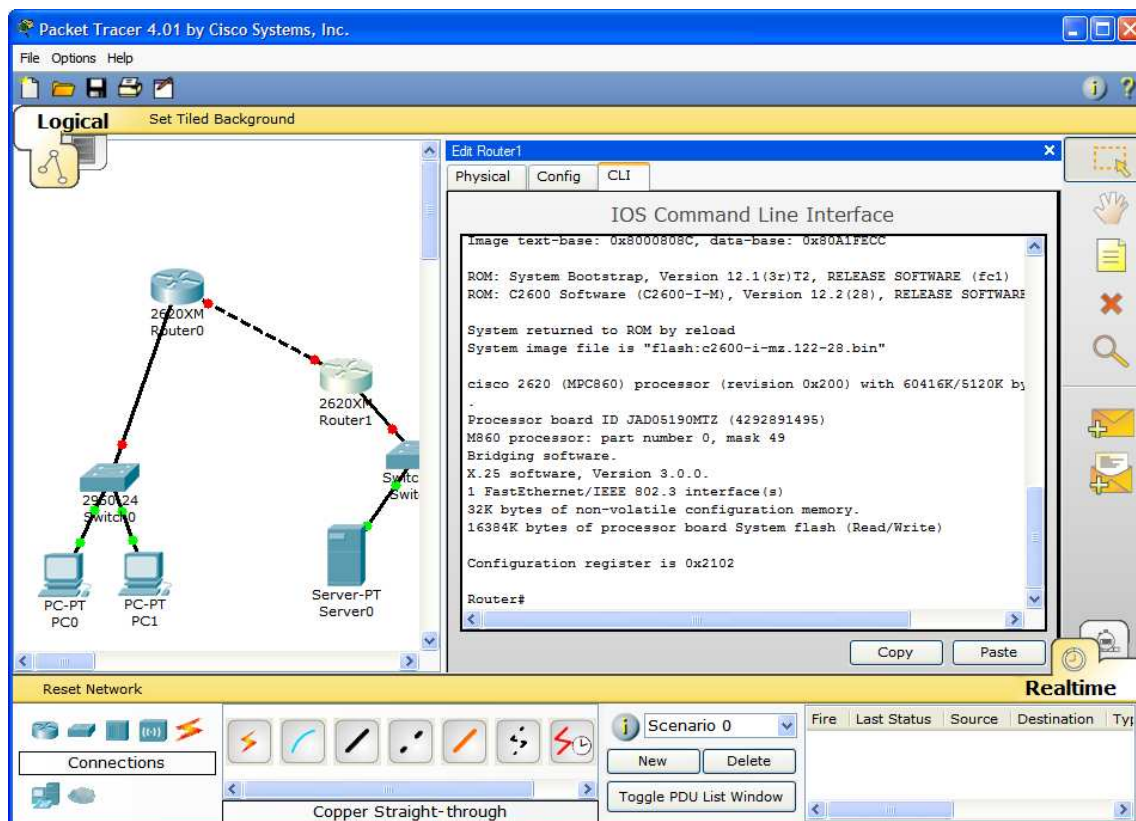




Obrázok 2.1: Ukážka prostredia programu Boson NetSim.

## 2.4. Packet Tracer

Packet Tracer je softvérovým simulátorom počítačových sietí priamo od spoločnosti Cisco. Je voľne stiahnuteľný pre účastníkov Cisco Networking Academy Program po prihlásení sa do systému. Simulátor pracuje pod OS Windows v grafickom móde. Umožňuje vytváranie fyzickej topológie siete, konfigurovanie rôznych sieťových prvkov v grafickom alebo v textovom móde. Je dobre ovládateľný a poskytuje veľa funkcií na simulovanie a testovanie vytvorených virtuálnych sietí. Packet Tracer je veľmi užitočnou aplikáciou, v ktorej sa dajú vytvoriť rôzne sieťové topológie, konfigurovať zariadenia, testovať prevádzku a pod. No bohužiaľ je obmedzená iba na určitú funkcionálnu, ktorú Packet Tracer v danej verzii poskytuje. Packet Tracer pri štarte zariadení nepoužíva skutočný IOS s jeho plnou funkcionálnou, ale používa len určitú obmedzenú skupinu funkcií, ktorú vývojári implementovali vo forme programového kódu. V Packet Tracer-y nie je možné zapnúť ladiace výpisy na niektoré funkcie, používať rozšírené vlastnosti ako VPN tunely, IPv6, *Frame Relay* prepínanie a pod.



Obrázok 2.2: Ukážka prostredia programu Packet Tracer.

## 2.5. Dynamips

Dynamips je emulátor hardvérovej platformy, založenej na MIPS procesoroch, konkrétne „enterprise platforme Cisco 7200“. Po spustení Dynamips aplikácie sa vytvorí virtuálne prostredie, ktoré bude emulovať skutočný hardvér smerovača rady 7200 (prípadne 3600). Pre reálne využitie emulovaného hardvéru je potrebný legálny operačný systém zariadenia - v tomto prípade ním bude operačný systém IOS pre smerovač Cisco 7200. Pri spustení Dynamips-u s cestou k IOS obrazu ako parametrom, sa spustí viac menej plnohodnotný smerovač Cisco 7200 s operačným systémom IOS. V konzole spustenej aplikácie sa zobrazí príkazový riadok smerovača, ktorý sprístupňuje skutočnú funkcionálnu a silu systému IOS (keďže už beží na skutočnom systéme). Cez rozšírené prepínače pri spúšťaní Dynamips je možné dokonca prepojiť sieťovú kartu vášho počítača s *Ethernet* portom virtuálneho smerovača, a tak napr. pripojiť virtuálnu 7200 do skutočnej topológie. Ako ďalšie hardvérové rozhrania, ktoré Dynamips umožňuje emulovať, patria už spomínané *Ethernet* porty, Sériové porty, ATM a iné. Nad Dynamips-om je napísaný *wrapper* Dynagen, ktorý veľmi jednoducho sprístupňuje konfiguráciu, vytváranie a sprístupňovanie virtuálnych smerovačov. Umožňuje vytvoriť napr. prostredie so 6

smerovačmi, navzájom prepojenými sériovými a *Ethernet* rozhraniami a zároveň s *Frame Relay* prepínačom.

```

TermServ#R1
Translating "R1"
Trying R1 (10.10.10.1, 2001)... Open
Connected to Dynanips UM "R1" (ID 0, type c3600) - Console port

Rack1R1#
Rack1R1#
Rack1R1#
TermServ#R2
Translating "R2"
Trying R2 (10.10.10.1, 2002)... Open
Connected to Dynanips UM "R2" (ID 1, type c3600) - Console port

Rack1R2#
Rack1R2#
Rack1R2#
TermServ#sh session
Conn Host          Address           Byte  Idle Conn Name
  1 R1              10.10.10.1       0     0 R1
 * 2 R2              10.10.10.1       0     0 R2
TermServ#_

```

Obrázok 2.3: Ukážka konzoly programu Dynamips.

## 2.6. VLAB

VLAB, teda virtuálne laboratórium, je program, ktorý vytvorili študenti fakulty v školskom roku 2007/2008. Je implementované pod operačným systémom Linux. Na emuláciu rôznych druhov Cisco zariadení používa emulátor Dynamips a rozhranie Dynagen.

### 2.6.1. Funkcionalita

Projekt má mať podľa používateľskej príručky nasledovné funkcionality:

- **Prihlásenie sa do systému** - Na prihlásenie sa do systému treba do webového prehliadača zadať adresu projektu VLAB. Po vyplnení správneho mena a hesla vám bude umožnený prístup.
- **Vytvorenie používateľa** - Pre vytvorenie nového používateľa klikneme na príslušnú položku a vyplníme požadované parametre (prihlasovacie meno,

heslo, skupina, meno, priezvisko a emailová adresa). Potvrdením vytvoríme nové konto pre používateľa.

- **Modifikácia údajov používateľa** - Otvoríme si položku pre zmenu údajov, zmeníme ich podľa potreby a potvrdíme.
- **Vytvorenie novej topológie** - Ak chceme vytvoriť novú topológiu otvoríme si príslušné okno a vyplníme požadované údaje (názov topológie, obrázok, počet smerovačov a konfiguráciu pre Dynagen).
- **Prehliadanie existujúcich topológií** - Na prehliadanie existujúcich topológií môžeme použiť filter alebo si nechať vypísať všetky topológie. V rámci výpisu si môžeme prezerať obrázok topológie a konfiguráciu (podľa potreby ju môžeme upraviť alebo aj zmazať).
- **Rezervácia času simulácie** - Pri rezervácii času si postupne vyberieme dátum a konkrétne hodiny, medzi ktorými chceme vykonávať simuláciu (simulácie). Svoj výber nakoniec potvrdíme.
- **Prehliadanie rezervácií** - Program umožňuje prezerať si všetky zadané rezervácie.
- **Spustenie simulácie** - Pre samotné spustenie simulácie je potrebné mať rezerváciu a v danom čase si zvolíme konkrétnu topológiu a konfiguráciu a spustíme samotnú simuláciu.

Podrobnejší popis funkcionalít programu spolu s názornými ukážkami nájdete v používateľskej príručke projektu VLAB z minulého roku [4].

Keďže systém registrácií cez web nie je funkčný, nedá sa program spustiť ako je uvedené v dokumentácii a ako nám to bolo prezentované. Testovali sme teda priame spustenie programu cez štartovacie skripty z konzoly, ktoré sú ekvivalentom iba s rozdielom, že výstup je priamo v okne konzoly a nie v okne prehliadača klienta. Vstupné parametre pre štartovacie skripty nezisťuje *proxy server*, ale je potrebné ich ručne vyselektovať z databázy. Pri takomto spôsobe spúšťania programu, keďže inú možnosť nemáme, sa nám simuláciu nepodarilo spustiť. Problém nastane vo fáze kedy sa Dynagen chce pripojiť na Dynamips. To sa mu nepodarí. Následne skript ukončí simuláciu, keďže sa vyskytne chyba. Možné príčiny sú nedostatočné práva prideleného konta, pod ktorým spúšťame simuláciu, keďže nepoužívame hlavnú vývojovú vetvu a ani konto správcu (*root*) ako hlavný vývojár Peter Péti. Respektíve príčinou môže byť, že samotné skripty sú napísané nekorektne, keďže dokážu pracovať iba s právami správcu.

## 2.6.2. Štartovacie skripty

Súčasnú riešenie spočíva v tom, že klient (Java *applet* konzola vo webovom prehliadači) sa pripojí na *proxy* server. *Proxy* server zabezpečí autentifikáciu užívateľa a skontroluje dobu, na akú má rezervovaný čas v systéme VLAB. Ak sa prihlási správny používateľ v správnu dobu (pre neho vyhradenú), *proxy* server akceptuje požiadavku a spustí požadovanú simuláciu.

Samotné spustenie simulácie a beh simulácie zabezpečujú 4 *bash* skripty: *startup.sh*, *startup\_user.sh*, *config.sh* a *shutdown.sh*.

Ako prvý sa spustí skript *startup.sh* *<PORT>* *<CONFID>* *<DURATION>*. Parameter **PORT** určuje port, cez ktorý sa klient prihlási na konzolu Dynamips-u. Od tohto portu sa odvodzujú aj porty jednotlivých zariadení v simulácii, na ktoré sa môže klient tiež pripojiť. Ak simulácia obsahuje 4 zariadenia, tak čísla ich portov budú: *PORT+1*, *PORT+2*, *PORT+3* a *PORT+4*.

Pri registrácii simulácie cez webové rozhranie má používateľ na výber z niekoľkých predvolených topológií. Topológia v databáze je uložená ako objekt typu „*lab*“. Dôležitá je tu položka *lab.configuration*, ktorá obsahuje popis topológie, aké zariadenia sú pre danú topológiu použité a ako sú navzájom poprepájané.

Príklad položky *lab.configuration* popisujúcej topológiu jedného smerovača s pripojeným počítačom:

```
model = 3640
autostart = True

[localhost:7200]
workingdir = /tmp/
port = 7200

[[3640]]
image = /opt/cisco/images/c3640-ik9o3s-mz.124-7.bin
ram = 128
rom = 4
nvram = 128

[[ROUTER R1PRAHA]]
model = 3640
console = 2001
f0/0 = host1
s1/0 = R1BA s1/0
idlepc = 0x6063900c

[[ROUTER R1BA]]
model = 3640
console = 2001
```

```
f0/0 = host2  
idlepc = 0x6063900c
```

Parameter **CONFID** je hodnota odkazujúca na *lab.id* v databáze. Hovorí o tom, aká topológia, teda aké konkrétne zariadenia sa majú emulovať v Dynamips-e.

Posledným parametrom je **DURATION**, čo je čas v sekundách, po ktorom sa ukončí Dynamips. Ak si používateľ zaregistroval 2 hodiny tak hodnota *DURATION* bude 7200.

Skript *startup.sh* po spustení skriptu *startup\_user.sh* čaká na návratovú hodnotu tohto skriptu, ak nastane chyba okamžite sa ukončí proces Dynamips-u, Dynagen-u a UML používateľských staníc. Ak sa Dynamips alebo Dynagen spustia korektne spustí sa skript *shutdown.sh* s parametrom, ktorý uvádza počet sekúnd, koľko bude simulácia trvať.

Úloha *startup.sh* overí či sa port *PORT* nevyužíva, ak áno vypíše chybové hlásenie a nepokračuje v spúšťaní simulácie. Pre spúšťanie simulácií je na serveri vytvorených 5 špeciálnych kont: *simu01*, *simu02*, *simu03*, *simu04* a *simu05*. Z toho vyplýva, že naraz môže byť spustených maximálne 5 simulácií. Skript *startup.sh* overuje, či niektoré konto nie je aktívne, ak nájde prvé ktoré sa nevyužíva prideli danej simulácii. Pridelenie spočíva v spustení skriptu *startup\_user.sh* pod právami *simu0x*.

```
sudo -H -u "$RUN_USER" $STARTUP_PATH $START_PORT $CONF_ID
```

Skript *startup\_user.sh* *<PORT>* *<CONFID>* (parametre sú prenesené zo skriptu *startup.sh*):

- spracuje konfiguráciu topológie z databázy a upraví ho pre konkrétnu simuláciu, položka *lab.configuration*,
- popripája koncové stanice k zariadeniam, ktoré majú na porty pripojene *hostX*,
- spustí VDE (virtuálny) prepínač,
- naštartuje jednotlivé UML koncové stanice (jednoduché linuxové virtuálne stoje),
- naštartuje *Hypervisor* (konzola programu Dynamips), teda spustí Dynamips a skontroluje či je aktívny na požadovanom porte *PORT*,

- spusti výsledný konfiguračný súbor do Dynagen-u,
- zároveň zapíše v databáze do konkrétnej simulácie, že je spustená, v čase kým je spustená a uloží aj jej konfiguráciu a čas spustenia.

Od tohto okamihu sa môže klient pripájať na jednotlivé zariadenia a ľubovoľne ich konfigurovať.

Skript *config.sh* sa volá zo skriptu *shutdown.sh* a tesne pred ukončením Dynamips-u načíta štartovací konfiguračný súbor každého smerovača (za predpokladu, že ho tam používateľ uložil) a uloží ho do databázy (položka *simulation.config*).

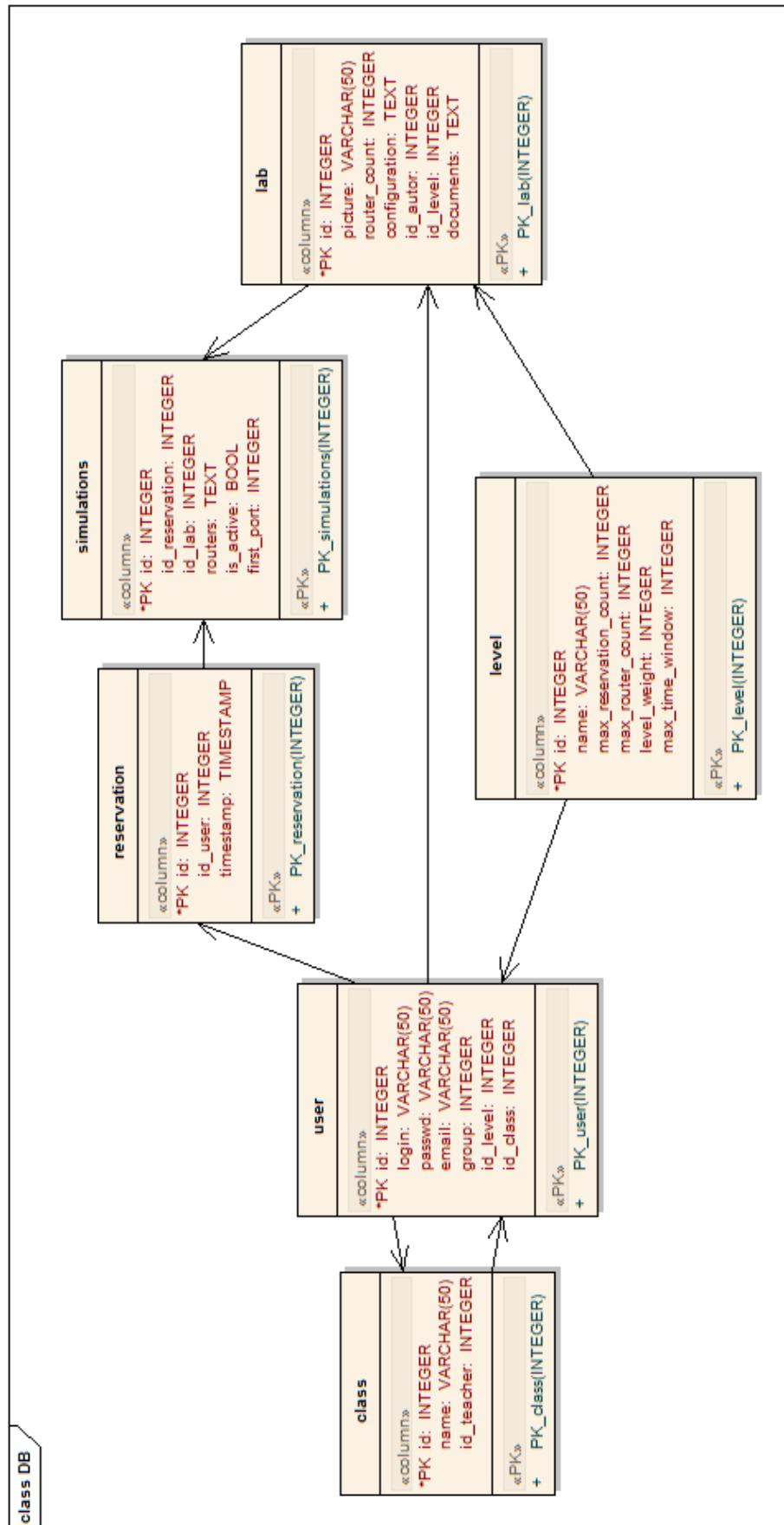
### 2.6.3. Databáza

System ukladá všetky svoje dáta do centrálnej databázy MySQL5. Tu sú uložené všetky údaje o systéme, používateľoch a simulovateľných topológiách.

Databáza obsahuje tabuľky [1]:

- **class** - Tabuľka obsahuje všetky virtuálne triedy v systéme, ktoré vytvára učiteľ.
- **user** - V tejto tabuľke sú uložení všetci používatelia systému, ktorí sú priradený do jednej z troch skupín a to administrátor, učiteľ alebo študent.
- **reservation** - Tu nájdeme rezervácie jednotlivých používateľov pre daný čas. Pri rezervovaní času sa v tabuľke vytvorí prislúchajúci záznam. Počet rezervácií je limitovaný. Je to z toho dôvodu, že systém je schopný súčasne emulovať len obmedzený počet zariadení.
- **simulation** - Táto tabuľka obsahuje potrebné položky pre vykonanie simulácie.
- **level** - Používatelia sú v systéme rozdelený do viacerých skupín (úrovní) podľa toho aké zložité a ako početné simulácie musia vykonávať.
- **lab** - Obsahuje všetky možné simulovateľné topológie. V tejto tabuľke sa nachádza položka „*picture*“, ktorá obsahuje konkrétny obrázok zvolenej topológie, tak aby bolo možné si ju priamo pozrieť.

Fyzický model databázy je uvedený na obrázku 2.4.



Obrázok 2.4: Fyzický model údajov [1].



### 2.6.4. Proxy

Hlavnou úlohou *proxy* servera je umožniť prihlásiť sa používateľovi na konzolu jednotlivých sieťových zariadení ako aj na hlavnú konzolu emulátora Dynamips. Ďalšou požiadavkou je aby bol danému používateľovi umožnený prístup len počas doby, ktorú má pridelenú podľa údajov v databáze. Zároveň musia byť spojenia medzi klientom a serverom zabezpečené proti odchyteniu komunikácie tretími stranami.

Pôvodný návrh počíta s vytvorením samostatného Java *applet-u*. Ten slúži ako konzola, cez ktorú sa používateľ vie prihlásiť na jednotlivé terminály. Táto aplikácia sa spúšťa cez ľubovoľný prehliadač. Návrh taktiež počíta iba s bezpečným pripojením sa medzi Java *applet-om* klienta a databázou. Komunikácia cez *proxy* server už nebola nijako zabezpečená.

Pôvodný návrh nepovažujeme za vhodné riešenie nakoľko použitie Java *applet-ov* cez internetové prehliadače sa z bezpečnostných dôvodov zvyknú zakazovať na klientskej strane. Zároveň platí že klient musí mať nainštalovanú podporu Java *applet-ov* ako takú.

### 2.6.5. Prepínač

Existujúce riešenie VLAB neumožňuje zapojiť a emulovať funkcie prepínača. Jediná možnosť je vytvoriť prepojenia iba cez klasický nemanážovateľný prepínač, čo je pre potreby reálnej emulácie siete nedostatočné.

Súčasná možnosť emulácie prepínača v Dynamips-e je emulovať smerovač s rozširujúcou kartou, ktorá obsahuje vstavaný manažovateľný prepínač. Táto karta však neumožňuje všetky funkcionality klasického manažovateľného prepínača. Jej funkcionality je závislá na type karty a verzii IOS-u.

### 2.6.6. Web rozhranie

Táto časť systému je predmetom diplomovej práce Bc. Petra Pétiho. Z tohto dôvodu sa tejto oblasti nebudeme venovať my a preto ho nebudeme ani analyzovať. Analýzu webového rozhrania môžete nájsť v dokumentácii k diplomovému projektu

spomínaného študenta. Pre potreby projektu však budeme úzko spolupracovať s týmto študentom, aby výsledok práce bol čo najpresnejší a spĺňal všetky požadované kritéria oboch strán.

## 3. Špecifikácia

Na základe analýzy existujúcich riešení sme sa rozhodli pokračovať v zdokonaľovaní projektu VLAB. V tejto časti dokumentácie uvedieme hlavne zmeny, ktoré chceme spraviť v existujúcom riešení virtuálne laboratórium. Sú to zmeny, ktoré skvalitnia a vylepšia výsledný produkt. Navyše sú tu spomenuté nové prvky, ktoré systém neobsahuje a chceme implementovať.

### 3.1. Zmeny vo VLAB

#### 3.1.1. Štartovacie skripty

Prvým krokom je prispôbenie rozhrania medzi *proxy* serverom a *skriptami*, aby sa program VLAB dal spustiť aj nezávisle bez *proxy* servera. V prípade keď vieme, že program VLAB dávame korektné vstupy, musí pracovať korektne.

V tejto časti chceme ďalej spraviť nasledovné úpravy:

- doladenie *skriptov*,
- doplnenie ladiacich výpisov,
- pridanie možnosti spustiť PIX/ASA,
- načítanie a ukladanie konfigurácie.

#### 3.1.2. Proxy

Chceme vytvoriť bezpečný *proxy* server, ktorý bude sám zabezpečovať bezpečnú komunikáciu medzi klientom a zvyšnými komponentmi. Zároveň chceme umožniť klientom pripájať sa cez známe programy a tým predísť problémom na klientskej strane. Nebude teda potrebné, aby mal klient nainštalovaný Java *applet*. *Proxy* server by mal spĺňať nasledovné požiadavky:

- komunikovať priamo s databázovým serverom,
- transparentne prenášať údaje medzi klientom a zariadeniami v danej simulácii.

## 3.2. Pridané prvky

### 3.2.1. Manažovateľný prepínač

Chceme pridať možnosť emulácie manažovateľného prepínača v daných topológiach, pomocou emulácie smerovača s rozširujúcou kartou obsahujúcu prepínač.

### 3.2.2. PIX Firewall

V našom riešení sa venujeme len emulácii PIX Firewall-u pomocou programu Pemu. Ten je integrovaný v najnovšej verzii programu ConfDynagen, ktorý už v našom systéme používame. Okrem samotného Dynamips-u bude potrebné spustiť v štartovacom skripte *startup.sh* aj Pemu server.

Momentálne ešte riešime problém so zaťažením procesora, nakoľko po spustení inštancie PIX-u zaťaží Pemu server procesor na 100%. Takto isto sa správajú aj smerovače v serveri Dynamips, avšak tam sa tento problém dá riešiť nastavením hodnoty *idlepc*. Toto v prípade Pemu nie je možné, takže budeme musieť použiť na obmedzenie buď nástroj operačného systému alebo externý program.

## 4. Návrh

Na základe špecifikácie uvádzame v tejto kapitole návrhy na riešenia špecifikovaných problémov a požiadaviek.

### 4.1. Štartovacie skripty

#### **Doladenie skriptov**

Niekedy sa nepodarí spustiť simuláciu, alebo sa simulácia korektne neukončí, treba ošetriť spracovávanie návratových hodnôt použitých príkazov.

#### **Doplnenie ladiacich výpisov**

Skripty okrem chybového hlásenia pochádzajúceho od volaných príkazov, samy neposkytujú žiadne informácie o chybe ktorá nastala, kde presne v kóde nastala a čo je príčina chyby. Skripty by mali poskytovať dostatočný popis chyby, ktorá nastane a zároveň by sa mali logovať s presným časovým označením.

#### **Pridanie možnosti spustiť PIX/ASA**

Skript, ktorý sa stará o vytvorenie konfiguračného súboru pre Dynagen a vytvára UML koncové stanice a virtuálny prepínač, bude rozšírený o schopnosť pridať do topológie PIX/ASA zariadenie.

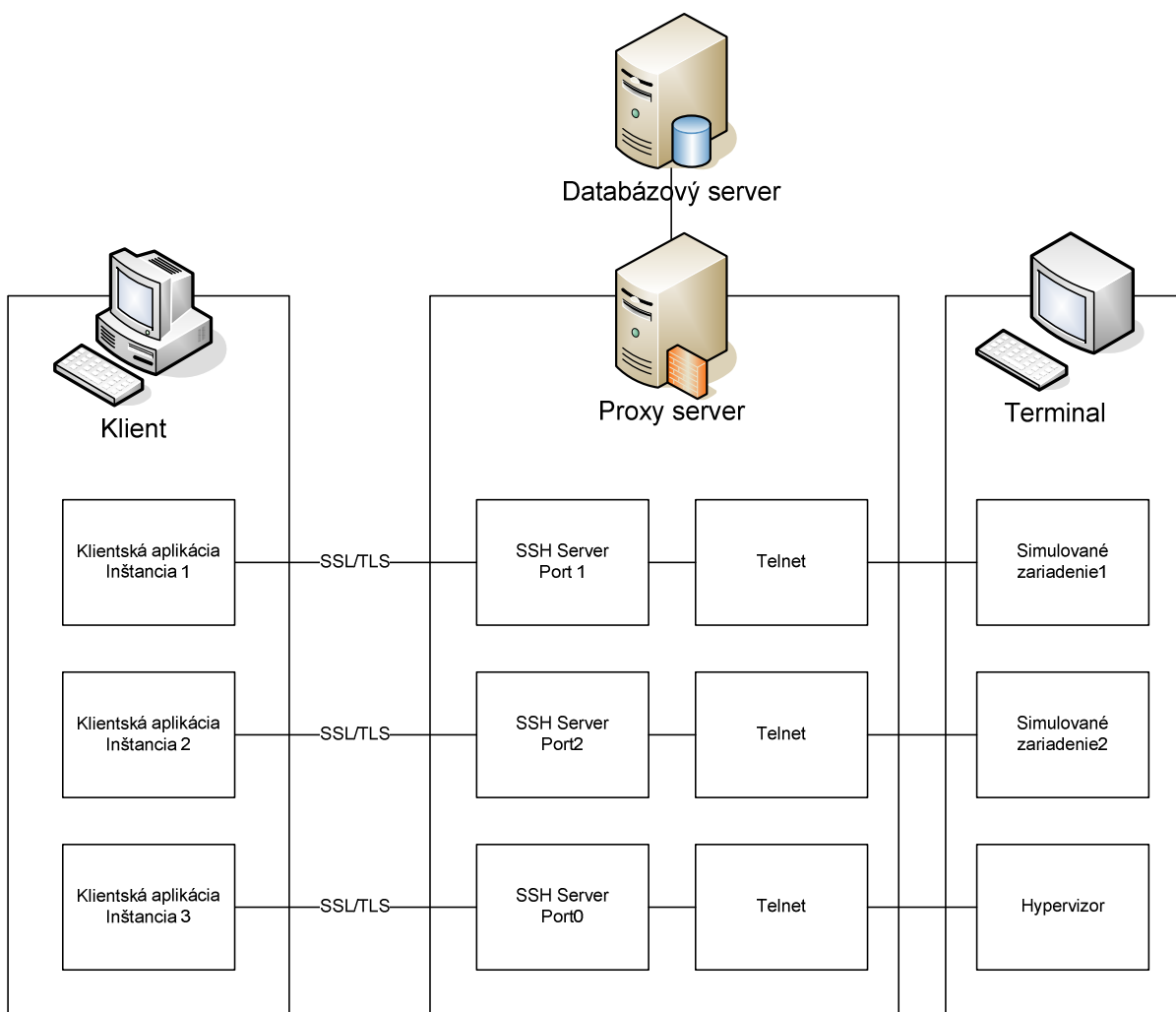
#### **Načítanie a ukladanie konfigurácie**

Počas simulácie bude možné uložiť štartovacie konfigurácie smerovačov. Pri ďalšej simulácii, ktorá bude pracovať s rovnakou topológiou si bude môcť používateľ načítať uloženú štartovaciu konfiguráciu a pokračovať tam, kde naposledy skončil.

## 4.2. Proxy server

*Proxy server* bude implementovaný ako jednoduchý SSH server, ktorý na autentifikáciu použije priamo databázu. Ako prístupové rozhranie sa následne použije výstup programu *Telnet*, ktorý bude pripojený výhradne na lokálny port daného sieťového zariadenia v simulácií.

Pre každé zariadenie v simulácií sa spustí zvlášť server, keďže je potrebné aby aj aplikácia na strane klienta vedela rozlíšiť kam sa pripája. Prípadne server ako taký sa môže spustiť iba raz, ale bude počúvať na viacerých portoch v závislosti od počtu terminálov (viď Obrázok 4.1).



Obrázok 4.1: Prepojenie *proxy* servera s ostatnými komponentmi.

### 4.3. Manažovateľný prepínač

Pri výbere topológie si používateľ zadá, že chce použiť manažovateľný prepínač, ktorý bude označený inak ako klasický prepínač. Je to z dôvodu, aby bolo jednoznačne definované, ktorý prepínač je klasický a ktorý prepínač je emulovaný cez smerovač s kartou. Následne systém vygeneruje potrebné konfiguračné súbory a pri spustení simulácie tento smerovač prednastaví tak, aby bolo jeho správanie čo najbližšie k reálnemu prepínaču.

### 4.4. PIX Firewall

Konkrétny emulovaný PIX bude vytvorený podobne ako smerovač, vygenerovaním súboru s topológiou pre Dynagen. Ten, ak nájde bežiaci Pemu server, vytvorí inštanciu virtuálneho PIX-u. Jeho prepojenie s ostatnými virtuálnymi zariadeniami funguje na rovnakom princípe ako v prípade smerovačov.

Spolu s *firewall-mi* budeme v topológiách orientovaných na sieťovú bezpečnosť spúšťať aj simulácie niektorých útokov. Na to nám slúži už používaná virtuálna koncová stanica s OS Linux. Tieto stanice boli v doterajšej implementácii virtuálneho laboratória používané na testovanie príkazom *ping*, prípadne na odskúšanie funkcie DHCP servera a pod. Na takúto virtuálnu stanicu doplníme program Nmap (prípadne nejaký ďalší s podobnou funkcionalitou), doplnený o potrebné skripty, ktoré požadovaný útok spustia.

### 4.5. Laboratórne cvičenia

Na to, aby mohol používateľ plne využívať náš projekt, pripravíme niekoľko topológií laboratórnych cvičení, ktoré vložíme do systému. Používateľ si vyberie cvičenie, ktoré si v danej chvíli potrebuje odskúšať. Laboratórne cvičenie bude obsahovať nasledovné položky:

- názov,
- konfiguráciu pre program Dynagen,
- topológiu (obrázok) zapojených zariadení.

Laboratórne cvičenia rozdelíme podľa potreby do niekoľkých úrovní na základe ich náročnosti pre študentov. Napríklad úrovne podľa predmetov, kde sa môže VLAB využívať:

- **PS2** - najnižšia úroveň, ktorá bude obsahovať jednoduchšie topológie. Cvičenia budú najmenej náročné na zaťaženie servera a budú teda využívať najmenej CPU a pamäte servera. Používateľ zaradený do tejto úrovne, bude môcť simulovať len tie laboratórne cvičenia priradené do úrovne PS2.
- **WAN** - stredná úroveň, ktorá bude obsahovať zložitejšie topológie ako predchádzajúca úroveň a teda budú viac zaťažovať server. Používateľ zaradený do tejto úrovne, bude môcť simulovať laboratórne cvičenia priradené k úrovni WAN a taktiež cvičenia zaradené do nižšej úrovne, teda PS2.
- **SEC** - najvyššia úroveň, ktorá bude obsahovať v topológiách PIX firewall, ktorý je najnáročnejší na zaťaženie CPU. Používateľ zaradený v tejto úrovni bude môcť simulovať všetky laboratórne cvičenia a to cvičenia úrovni Security, WAN a PS2.

Možnosť vytvárať a meniť laboratórne cvičenia bude mať pravdepodobne len administrátor a učiteľ. Je to z toho dôvodu, aby sme predišli vytvoreniu chybných laboratórných cvičení. Ak bude chcieť študent pridať zadanie, pošle ho svojmu vyučujúcemu, ten ho skontroluje a vloží do databázy.



## 5. Implementácia

V nasledovnej kapitole nájdete uvedené informácie o tom ako sme implementovali jednotlivé časti projektu. V prípade, že sa implementácia líši od návrhu, uvádzame aj vysvetlenie prečo sme sa odklonili od pôvodnej myšlienky.

### 5.1. Štartovacie skripty

**Makefile** vytvorí:

- spustiteľný súbor *startup*, ktorý zabezpečuje spúšťanie skriptu *startup.sh* a všetkých skriptov, ktoré sú spustené zo skriptu *startup.sh* s právami daného používateľa. Toto riešenie je kvôli fungovaniu daného *shell-u*, kedy nie je možné spustiť skript *startup.sh* a jeho vnorené skripty pod požadovaným používateľom.
- spustiteľný súbor *dynagen\_wrapper*, ktorý spúšťa program Dynagen. Dynagen neobsahuje prepínač, ktorý mu umožní bežať na pozadí. Preto sa spúšťa prostredníctvom súboru *dynagen\_wrapper*, ktorý to umožní. Beh na pozadí je potrebný pre spustenie programu Dynagen v skripte, inak by sa muselo čakať, kým Dynagen skončí.
- spustiteľný súbor *shell*, ktorý predstavuje nami vytvorený *shell* pre CLI v systéme. Tento *shell* spustí Telnet na lokálny port. Nie je možné z neho volať žiadne systémové volania ani spúšťať príkazy v systéme.

**headers.sh** - obsahuje premenné a cesty pre skripty (obdoba hlavičkového súboru). Taktiež obsahuje 2 funkcie, ktoré nájdú prvý voľný port, na ktorom sa spustí nová simulácia a nasledujúce porty pre jednotlivé zariadenia v simulácii.

**startup.sh** - nájde voľný port pre Dynamips hypervisor. Tento port zrežazí s menom používateľa a vytvorí v systéme nového používateľa, pod ktorým bude bežať celá simulácia. Napríklad *peter\_7200*.

Pod týmto novým používateľom sa spustia nasledovné skripty:

- **startup\_user.sh** - spustí Dynamips, Dynagen, virtuálne zariadenia (*hosts*), prípadne Pemu a PIX firewall podľa danej konfigurácie simulácie.
- **create\_users.sh** - vytvorí v systéme používateľov pre každé zariadenie v simulácii. Používateľovi VLAB-u budú následne zobrazené tieto prihlasovacie mená (*logins*) do systému v tvare *adresa\_servera, meno\_zariadenia\_v\_simulácii: login*. Napríklad : *vlab.fiit.stuaba.sk:22 Router\_R1: peter\_7201*. Ako heslo sa používa používateľovo heslo do systému VLAB.

**shutdown.sh** - po čase vyhradenom pre danú simuláciu vypne všetky emulované smerovače v programe Dynamips, zruší virtuálne hosty a všetky ostatné procesy, ktoré boli spustené pod pôvodným používateľom *peter\_7200*. Taktiež zruší používateľské kontá do systému pre jednotlivé zariadenia. Skript obsahuje nasledujúce parametre:

- CONF\_ID SIMU\_DURATINON LOGIN PASSWORD [TEST],
- CONF\_ID – číslo simulácie z databázy,
- SIMU\_DURATINON – dĺžka trvania simulácie v sekundách,
- LOGIN – meno používateľa vo VLAB-e,
- PASSWORD – heslo používateľa do VLAB-u,
- TEST – voliteľný parameter, pre testovanie konfiguračných súborov pre Dynagen. Skript vráti hodnotu 0 ak sa podarilo s daným konfiguračným súborom spustiť simuláciu bez chyby, alebo hodnotu 1, ak nastala počas spúšťania simulácie chyba. Toto však nezaručuje na 100%, že chyba bola spôsobená chybným konfiguračným súborom.

**startup\_user.sh** - skript nájde voľný port a spustí na ňom Dynamips hypervisor. Na najbližších voľných portoch postupne spustí PEMU, UML zariadenia (*hosts*), smerovače, prepínače, PIX *firewall-y*. Skript načíta pôvodný konfiguračný súbor programu Dynagen a na základe kľúčových slov: „router“, „host“, „FW“, „rswitch“, vytvorí pre tieto zariadenia ich virtuálnu obdobu a pripojí na Dynamips cez VDE prepínač (prepája UDP porty na *localhost*).

Po úspešnom spustení všetkých zariadení v simulácii a ich prepojení sa do databázy uloží XML popis danej simulácie, vrátane zariadení a ich portov. Z tohto XML súboru sa používateľovi propaguje informácia o prihlasovacích údajoch pre jednotlivé zariadenia v simulácii.

**create\_users.sh** - skript vytvorí pre každé zariadenie v simulácii nového používateľa v systéme s heslom rovnakým ako má používateľ do VLAB-u. Ako prednastavený *shell* pre týchto používateľov je náš *shell*, ktorý spustí priamo *Telnet* na port na *localhost*, kde beží zariadenie prislúchajúce danému používateľovi.

**shutdown.sh** - skript sa uspí na čas vyhradení na beh simulácie. Po tomto čase vymaže zo systému používateľa, pod ktorým bola spustená simulácia (v našom prípade *peter\_7200*) a takisto aj používateľov pre jednotlivé zariadenia v simulácii. Takisto zruší všetky procesy spustené pod používateľom *peter\_7200*, teda *Dynamips*, *UML hosts*, *Pemu*, *vde\_switch* prípadne *PIX firewall-y*. Pred vypnutím zariadení v simulácii sa spustí skript *config.sh*.

**config.sh** - z bežiacich smerovačov v programe *Dynamips* sa zoberie ich konfigurácia (*startup\_config*) a uloží sa do databázy. Používateľ tak má možnosť ukončiť simuláciu a nestratiť konfiguráciu. Tú si môže kedykoľvek v systéme pozrieť.

## 5.2. Proxy server

Pri naštudovaní viacerých rôznych implementácií SSH serverov sme zistili, že úprava týchto zdrojových kódov by bola nad časový rámec tímového projektu. Preto sme sa rozhodli použiť už existujúce riešenia a prepojiť ich medzi sebou. Základne časti *proxy* servera sú:

- SSH server,
- *Telnet* klientska aplikácia,
- *Shell* skript.

### 5.2.1. SSH server

Ako SSH server sme použili najčastejšie používaný SSH server *OpenSSH*. Do zdrojových kódov nebolo nutné zasahovať. V nastaveniach SSH serveru bolo potrebné umožniť pripájanie sa dynamicky generovaných používateľov. Z tohto dôvodu je možné použiť v podstate ľubovoľný SSH server.

SSH server v našom prípade slúži na bezpečné pripojenie sa medzi používateľom a samotným *proxy* serverom.

### 5.2.2. Telnet klientska aplikácia

Ďalšou súčasťou *proxy* servera je nevyhnutne podporovanie protokolu *Telnet* pomocou ľubovoľnej *Telnet* klientskej aplikácie. V našom prípade bola použitá štandardná OpenBSD *Telnet* aplikácia. Systém umožňuje použitie aj inej klientskej aplikácie no je nevyhnutné vykonať potrebné úpravy v zdrojových kódach *Shell* skriptu. *Telnet* aplikácia slúži na komunikáciu medzi *proxy* serverom a jednotlivými zariadeniami v danej simulácii. Tieto spojenia sa vytvárajú až na základe inicializácie SSH spojenia z SSH servera. Naše riešenie predpokladá umiestnenie inštancie Dynamips na tom istom serveri ako *proxy* server alebo na vzdialenom serveri, kde spojenie medzi *proxy* serverom a serverom Dynamips je bezpečné a nie je potrebné ho opätovne zabezpečovať.

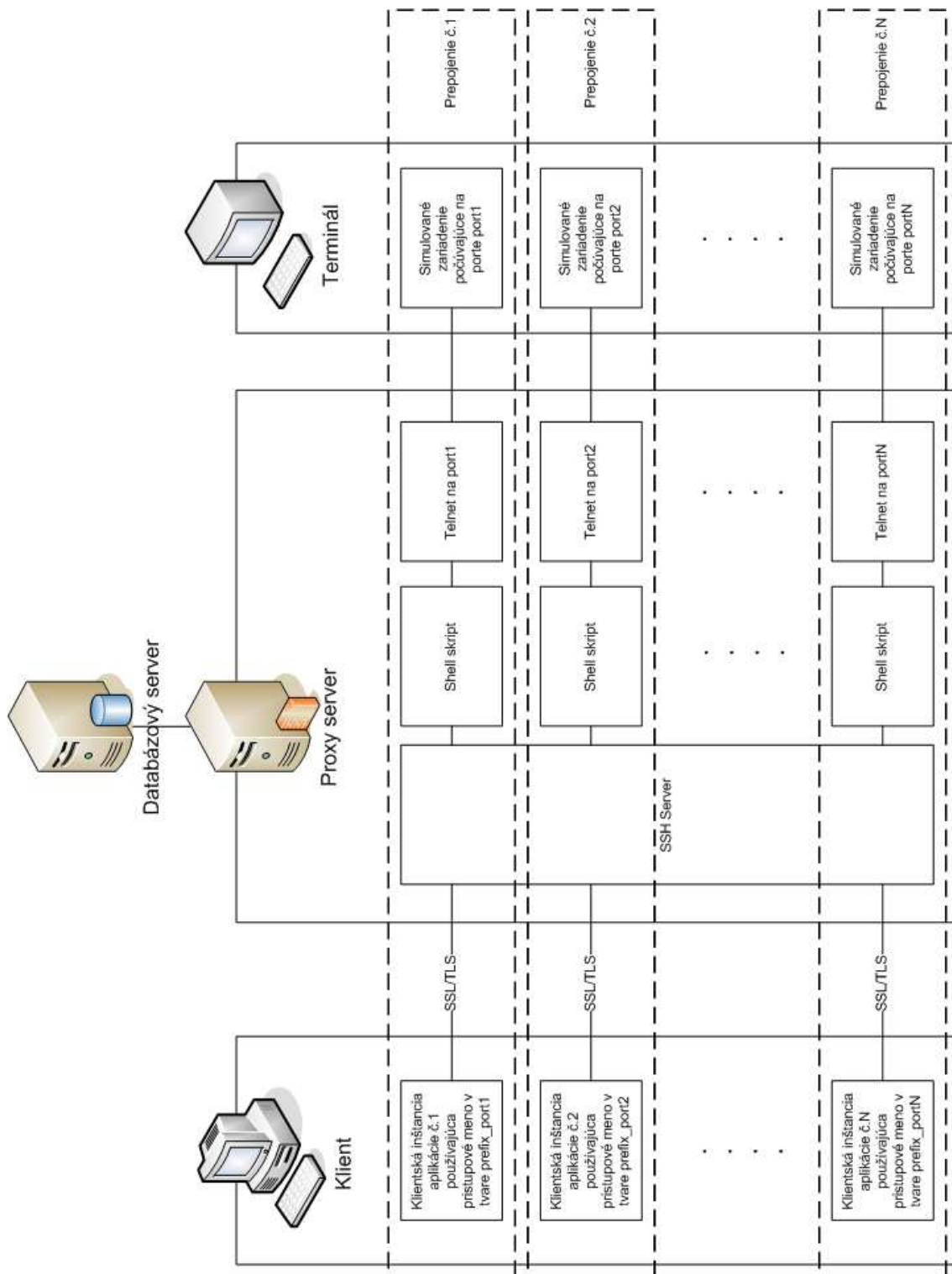
### 5.2.3. Shell skript

*Shell* skript slúži na prepojenie SSH servera a *Telnet* spojenia a tým vytvára transparentné prepojenie medzi používateľom a samotnými zariadeniami v simulácii. Mapovanie jednotlivých používateľov na simulované zariadenia je realizované prostredníctvom nastavenia parametru „*shell*“ na cestu k nášmu *Shell* skriptu. Každý používateľ ma napevno priradené simulované zariadenia, na ktoré sa môže prihlásiť prostredníctvom dynamických používateľských mien v tvare *prefix\_port*. *Shell* skript pri svojej inicializácii jednoducho nájde posledný výskyt znaku „\_“ a zvyšok reťazca je považovaný za port na ktorom počúva simulované zariadenie programu Dynamips. V prípade ak sa nepodarí otvoriť dané *Telnet* spojenie alebo sa nejakým spôsobom preruší, dochádza k spätnému ukončeniu *Shell* skriptu a tým pádom aj k ukončeniu SSH spojenia. Tým je zabezpečená možnosť, aby sa používateľ nedokázal prihlásiť na *proxy* server ako reálny používateľ alebo správca. Títo dynamickí používatelia sa po ukončení celej simulácie používateľom cez web rozhranie alebo vypršaní časového limitu automaticky odstránia, čo taktiež prispieva k bezpečnosti.

Pre umožnenie spustenia *Shell* skriptu je potrebné ho pridať do zoznamu *Shell* skriptov, čo je automaticky realizované pri prvotnej inštalácii celého systému.

### 5.2.4. Prepojenie celého systému proxy servera

Výsledné prepojenie medzi používateľom a simulovaným zariadením vyzerá nasledovne:



Obrázok 5.2: Prepojenie *proxy* servera s ostatnými komponentmi.

## 5.3. Manažovateľný prepínač

Ďalšou z úloh nášho projektu bola implementácia fungujúceho manažovateľného sieťového prepínača. Ideálne by bolo dosiahnuť tento cieľ pomocou emulácie reálneho hardvéru. Bohužiaľ, na rozdiel od smerovačov, sieťové prepínače používajú hardvérové ASIC obvody, ktorých štruktúra je tajná a dala by sa získať len reverzným inžinierstvom. Tento proces by bol veľmi náročný na čas a vedomosti.

Najlepším dostupným riešením je emulácia smerovača so zabudovaným modulom NM-16SW. Tento modul slúži na sprístupnenie prepínacích funkcií modulárnym smerovačom. Obsahuje 16 *Ethernet* portov s rýchlosťou 10/100 Mbps. Ak je takýto modul vložený do smerovača, je možné zadávať príkazy špecifické pre prepínač. Tento modul podporuje funkcie ako VLAN, VTP, STP, „*trunking*“, ktoré sú potrebné pre študentov používajúcich náš projekt. Bohužiaľ niektoré funkcie, menovite RSTP, MSTP, „*port security*“, tento modul nepodporuje. Uvažovali sme aj o zakomponovaní reálne fungujúceho softvérového prepínača, naprogramovať ho však so všetkými funkciami modulu NM-16SW spolu s nepodporovanými by zabralo veľké množstvo času, a navyše by nebola zabezpečená dokonalá spolupráca s programom Dynagen.

Samotná implementácia spočívala v upravení spúšťacích skriptov, aby používateľ vkladajúci konfiguráciu simulácie nemusel NM-16SW moduly pridávať ručne. Stačí, ak bude v konfigurácii miesto položky `[[router]]` zapísaná položka `[[rswitch]]`. Vtedy skript doplní parametre špecifikujúce model smerovača 3640 a vloženú kartu NM-16SW. Zadné porty, ku ktorým používateľ v konfigurácii pripájal iné zariadenia zo simulovanej topológie, ostali zachované.

Chceli sme však dosiahnuť, aby sa takto nakonfigurovaný smerovač z pohľadu používateľa tváril ako prepínač. Vytvorili sme teda štartovaciu konfiguráciu smerovača, v ktorej zmeníme jeho meno z „Router“ na „Switch“ (ako to je na hardvérových Cisco prepínačoch) a vypneme pomocou konfiguračného príkazu „*no ip routing*“ smerovanie. Tým zabezpečíme, že sa bude takýto smerovač od začiatku správať ako prepínač druhej vrstvy. Konfigurácia je uložená v jednom súbore na serveri, nakoľko je spoločná pre všetky topológie používajúce takýto prepínač. Jej priradenie ku konkrétnemu smerovaču vykonáva náš skript. Používateľ takto na prvý pohľad ani nezistí, že je pripojený na smerovač a nie na prepínač.

## 5.4. PIX Firewall

Implementácia *firewall-u* je v najnovšej verzii programu Dynagen už priamo zahrnutá pomocou emulátora Pemu. Tento emulátor vznikol modifikáciou univerzálneho emulátora Qemu, ktorý umožňuje emulovať rôzne hardvérové architektúry. Pemu je prispôbený pre zariadenie PIX od firmy Cisco (prípadne aj pre zariadenie ASA) a podporuje všetky verzie operačného systému PIX-u (s ASA je problém, pretože najnovšie verzie s Pemu nefungujú). Samotné spustenie spočíva v doplnení riadkov pre *firewall* do spúšťacieho *.net* súboru pre Dynagen. Keď bude Dynagen tento súbor pri spúšťaní čítať, v momente keď narazí na riadok oznamujúci použitie *firewall-u*, spojí sa s inštanciou Pemu servera (Tú bolo potrebné spustiť dopredu. Chová sa podobne ako Dynamips hypervisor, teda beží v pozadí a čaká, kým sa naň napojí Dynagen a spustí konkrétne inštancie *firewall-u*) na porte 10525 a predá mu parametre PIX/ASA *firewall-u*, ktorý sa má vytvoriť. Tieto parametre obsahujú názov súboru „*image*“, počet rozhraní typu Ethernet, UDP porty, ktorými sa tieto rozhrania pripoja k serveru Dynamips, číslo portu pre konzolový prístup, veľkosť pamäti RAM a názov súboru slúžiaceho ako *Flash* pamäť.

Iným riešením je priamo použitie originálneho emulátora Qemu. Ten má oveľa zložitejšie nastavovanie spúšťacích parametrov a je potrebné ručne priraďovať čísla portov pre UDP tunely, pomocou ktorých sa Qemu spojí s programom Dynamips. Na druhej strane je výhodou Qemu podpora zariadenia ASA so všetkými momentálne dostupnými verziami jeho operačného systému.

Pre náš projekt sme vybrali prvý prípad, teda spustenie pomocou Pemu hypervisora a doplnenie *.net* súboru o riadky s parametrami zariadenia *firewall*. Viedlo nás k tomu jednoduchšie vytváranie UDP tunelov (to sa v tomto prípade vykonáva automaticky), keďže pri použití Qemu by sme museli pozorne vytvoriť ďalšie skripty, ktoré by na úrovni UDP tunelov prepojili Dynamips s Qemu (podobne ako to je v našom projekte riešené pri simuláciách využívajúcich UML koncové stanice, s čím sme mali počas vývoja neustále problémy). Jedinou nevýhodou tak ostala nemožnosť emulácie zariadenia ASA v najnovšej verzii softvéru. To však samo o sebe oproti PIX-u neprináša veľa výhod, pretože hlavným vylepšením je použitie zásuvných hardvérových modulov, ktoré tak či tak nie je možné emulovať. Jediná funkcia, ktorá by mohla byť na ASA v porovnaní s PIX užitočná je WebVPN, teda vytváranie VPN spojení bez softvérového klienta. Náš projekt je však určený hlavne na uľahčenie výučby základov práce so sieťovou bezpečnosťou, kde táto vlastnosť nie je podľa nás zvlášť potrebná.

Samotná implementácia spočívala v doplnení spúšťacích skriptov o kontrolu prítomnosti *firewall-u* v štartovacom *.net* súbore. Ak sa tam *firewall* nachádza, skript musí spustiť jedného hypervisoru pre Pemu server. Tu sme narazili na veľmi veľký problém. Dynagen ani Pemu totiž nepodporujú zmenu portu, na ktorom Pemu bude bežať, resp. na ktorom sa bude Dynagen pripájať. Vždy sa použil štandardný port 10525, čo znemožňovalo spustenie dvoch a viac simulácií využívajúcich Pemu. Bolo teda nutné upraviť zdrojové súbory oboch programov, kde bola doplnená možnosť zadania parametra pri spustení Pemu servera, ktorý špecifikuje číslo portu pre pripojenie. Do *.net* súborov je pomocou skriptu automaticky doplnený parameter „*fwport*“, ktorý určuje počúvajúci port spusteného Pemu servera. Ako všetky porty v našom projekte, aj tento sa určuje automaticky. Následne je s hodnotou tohto parametra spustený Pemu server. Tak je zabezpečené, že Dynagen a Pemu budú komunikovať na tom istom porte.

Pemu nepodporuje zníženie záťaže CPU pri simulácii (*idlepc* pri programe Dynagen) a preto vyťažší procesor vždy na 100%. To je pre náš projekt nepoužiteľné, preto bolo do spúšťacieho skriptu pre Pemu doplnené po spustení samotného PIX *firewall-u* aj následné spustenie programu **cpulimit**, ktorý dostane ako spúšťací parameter názov vykonateľného súboru a počet %, ktorými môže tento súbor zaťažiť procesor počítača. Obmedzenie Pemu sme nastavili na 10%, pri menších hodnotách už bolo konfigurovanie po pripojení na konzolu značne spomalené. Nie je to síce tak „elegantné“ riešenie, ako používa Dynagen, avšak emulácia PIX/ASA momentálne lepšie zníženie zaťaženia procesora neumožňuje.

## 5.5. Laboratórne cvičenia

Laboratórne cvičenia sme rozdelili do 4 úrovní tak, ako to vidno na nasledovnom obrázku.

Name	Max. reservation count	Max. memory usage [ MB ]	Max. cpu usage [ % ]	Max. time window	
PS2	6	512	21	4	Edit   Delete   Up   Down
WAN	4	800	35	6	Edit   Delete   Up   Down
CCNA	4	1024	40	8	Edit   Delete   Up   Down
SEC	2	1024	43	10	Edit   Delete   Up   Down

Obrázok 5.3: Úrovně laboratorných cvičení.



Tabuľka je rozdelená do piatich stĺpcov:

- **Name** - obsahuje názov úrovne podľa toho, aké laboratórne cvičenia obsahuje. PS2 a WAN sú predmety, na ktorých sa dané laboratórne cvičenia precvičujú. Úroveň CCNA obsahuje všetky cvičenia, s ktorými sa študent stretne pri absolvovaní všetkých štyroch semestrov. Posledná úroveň je SEC, kde sa nachádzajú cvičenia zamerané na bezpečnosť. Vzhľadom na to, že obsahujú PIX *firewall*, sú náročné na CPU.
- **Max. reservation count** - udáva maximálny počet rezervácií jedného používateľa danej úrovne. Napríklad úroveň PS2 má túto hodnotu nastavenú na 4. Pre používateľa v tejto úrovni to znamená, že si môže vytvoriť 4 nezávislé rezervácie. Ak by chcel 5, systém mu to nedovolí a musí počkať kým simuláciu ukončí, alebo si bude musieť zmazať jednu z existujúcich. Toto zabezpečuje systém pred tým, aby si jeden používateľ vytvoril rezervácie na celý týždeň a bránil tým používať VLAB iným používateľom.
- **Max. memory usage [MB]** - udáva maximálne využitie pamäte servera pre daný level. Teda ak sú v danej úrovni cvičenia s rôznymi hodnotami záťaže pamäte, tak túto hodnotu nastavíme ako najvyššiu spomedzi nich.
- **Max. cpu usage [%]** - udáva maximálne využitie procesora servera pre daný level. Zadávame ju podobne ako pri využití pamäte. Celkové využitie procesora je nastavené na maximálne 85%. To znamená, že súčasne môžu systém využívať dvaja používatelia úrovne CCNA alebo WAN, alebo štyria používatelia úrovne PS2, prípadne dvaja používatelia úrovne PS2 a jeden používateľ vyššej úrovne a podobne.
- **Max. time window** - udáva maximálnu dĺžku rezervácie. Napríklad používateľ úrovne WAN si môže naraz rezervovať maximálne 6 hodín. Tento parameter sa zvyšuje spolu s úrovňou tak, aby si používateľ stihol odskúšať čo potrebuje. Je teda logické, že používateľ s vyššou úrovňou bude potrebovať viac času na náročnejšie cvičenia. Používateľ si však môže rezervovať aj kratší časový úsek podľa potreby.

V nasledovnom obrázku vidno zoznam všetkých laboratórnych cvičení, ktoré sú momentálne uložené v databáze. Každé cvičenie má svoj názov, autora, príslušnú úroveň a prepočítané hodnoty maximálneho využitia pamäte a procesora servera.

Name	Memory usage [ MB ]	CPU usage [ % ]	Author	Level	
Router-Host	320	14	PiT	WAN	Show picture   Show config   Edit   Delete
Basic BGP	640	25	admin Admin	WAN	Show picture   Show config   Edit   Delete
Frame Relay - set 1	512	20	admin Admin	WAN	Show picture   Show config   Edit   Delete
Frame Relay - set 2	512	20	admin Admin	WAN	Show picture   Show config   Edit   Delete
PPP	448	19	admin Admin	WAN	Show picture   Show config   Edit   Delete
Static and dynamic configuration	320	14	admin Admin	PS2	Show picture   Show config   Edit   Delete
Pokročilé smerovanie	480	21	admin Admin	CCNA	Show picture   Show config   Edit   Delete
BGP a iné smerovacie protokoly	640	28	admin Admin	CCNA	Show picture   Show config   Edit   Delete
Smerovanie v zložitej sieťovej topológii	768	33	admin Admin	CCNA	Show picture   Show config   Edit   Delete
Jednoduchy PIX lab	256	25	admin Admin	SEC	Show picture   Show config   Edit   Delete
Pokročily PIX lab	256	40	admin Admin	SEC	Show picture   Show config   Edit   Delete
IBGP and EBGP	416	17	admin Admin	WAN	Show picture   Show config   Edit   Delete
DHCP	160	7	admin Admin	WAN	Show picture   Show config   Edit   Delete
CCNA3_lab5_cs1	480	36	admin Admin	CCNA	Show picture   Show config   Edit   Delete
CCNA3_lab5_cs2	576	39	admin Admin	CCNA	Show picture   Show config   Edit   Delete
CCNA3_lab5_STProotBridge	256	23	admin Admin	CCNA	Show picture   Show config   Edit   Delete
CCNA3_lab7	64	14	PiT	CCNA	Show picture   Show config   Edit   Delete
Basic Router Configuration	160	7	admin Admin	PS2	Show picture   Show config   Edit   Delete

Obrázok 5.4: Zoznam laboratórných cvičení.

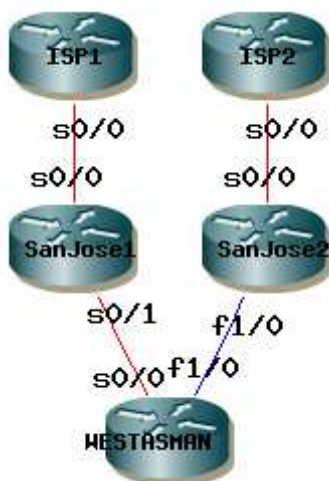
Preto, aby sme vedeli vypočítať, koľko ktoré cvičenie môže maximálne zaťažiť pamäť a procesor serveru, stanovili sme si tieto hodnoty pre jednotlivé zariadenia a to nasledovne:

zariadenie	vyťaženie pamäte	vyťaženie procesora
smerovač	160 MB	5 %
počítač	0 MB	2 %
prepínač	0 MB	0 %
prepínač <i>Frame Relay</i>	0 MB	0 %
manažovateľný prepínač	160 MB	5 %
PIX <i>firewall</i>	256 MB	15 %

Hodnoty sme zisťovali zo systému priamo po spustení jednoduchých topológií. Zaťaženie procesora je mierne vyššie ako ukazoval systém a to z toho dôvodu, aby nedošlo náhodou k situácii, keď simulácia zoberie viac prostriedkov, ako má vyhradené. Táto skutočnosť by mohla ovplyvniť všetky simulácie, ktoré v danom čase prebiehajú.

Ako príklad uvádzame jedno laboratórne cvičenie, ostatné si môžete prezrieť priamo na stránke projektu [vlab.fiit.stuba.sk](http://vlab.fiit.stuba.sk) v časti **LAB**:

- **názov:** „Basic BGP“
- **topológia:** Samotný obrázok topológie je generovaný programom Graphviz na základe vlozenej konfigurácie.



Obrázok 5.5: Topológia cvičenia „Basic BGP“.

- **konfigurácia:**

```

autostart = True
[localhost]

[[3640]]
image = /opt/cisco/images/c3640-ik9s-mz.124-16.bin
idlepc = 0x605d60a8
ram = 128
rom = 4
nvram = 128
sparsemem = True
ghostios = True
chassis = 3640

[[router WESTASMAN]]
model = 3640
console = 2006
slot0 = NM-4T
s0/0 = SanJose1 s0/1
f1/0 = SanJose2 f1/0

```

```
[[router SanJose1]]
model = 3640
console = 2007
slot0 = NM-4T
s0/0 = ISP1 s0/0
```

```
[[router SanJose2]]
model = 3640
console = 2008
slot0 = NM-4T
s0/0 = ISP2 s0/0
```

```
[[router ISP1]]
model = 3640
console = 2009
slot0 = NM-4T
```

```
[[router ISP2]]
model = 3640
console = 2010
slot0 = NM-4T
```

- **popis** - obsahuje krátky popis cvičenia a odkaz na súbor, ktorý presne popisuje kroky, ktoré môže používateľ robiť vo vybranom cvičení, podobne ako je to v laboratórnych cvičeniach Cisco akadémie.

## 6. Testovanie a overenie riešenia

### 6.1. Proxy server

#### 6.1.1. Overenie bezpečnosti použitých aplikácií

Overovanie bolo realizované iba pre *Shell* skript, kde sme sa snažili zaútočiť na jeho bezpečnosť a tak získať ďalšie práva. Tieto testy však skončili s úspešne, keďže sa nám nepodarilo nijako prelomiť daný relatívne jednoduchý kód.

Bezpečnosť ostatných súčastí nebola testovaná z dôvodu, že ich implementácia je štandardná a vyvíjaná inými autormi. Preto odporúčame použiť výhradne stabilné verzie daných súčastí.

#### 6.1.2. Testovanie záťaže servera

Testovanie záťaže bolo uskutočnené pomocou generovania spojení simulujúcich pripojenie sa rozličných používateľov v jednom čase s cieľom zistiť maximálny počet spojení. Ukázalo sa, že systém je dostatočne stabilný a nepadol ani raz z dôvodu maximálneho počtu spojení. Maximálny počet spojení, ktoré sa nám podarilo vygenerovať bol viac ako 60000 simultánnych spojení, čo zodpovedá nastaveniu celého systému. Z nášho pohľadu je takýto limit viac ako postačujúci a počet reálnych spojení by nikdy nemal presiahnuť túto hranicu.

#### 6.1.3. Overenie bezpečnosti niektorých z aplikácií

Pri tomto teste sme sa snažili zistiť, či je možné obísť pripojenie cez všetky súčasti *proxy* servera a tak získať nepovolené práva. Zistilo sa, že Dynamips (hoci v dokumentácií sa uvádza že vie počúvať len na niektorých IP adresách) sa správa neštandardne a reálne počúva na všetkých IP adresách pridelených systému, čo

umožňuje používateľovi použiť priamo aplikáciu *Telnet* na pripojenie sa na dané simulované zariadenie, čo je v rozpore s našimi požiadavkami. Keďže Dynamips je ešte stále vo vývoji a zasahovať do zdrojových kódov nebolo našim cieľom, navrhli sme použitie bezpečnostnej brány, ktorá bude pokus o pripojenie priamo na zariadenie cez *Telnet* filtrovať. Realizáciu tejto bezpečnostnej brány nechávame na administrátora systému, keďže nevieme predpokladať čo všetko je potrebné filtrovať a aké ďalšie služby budú bežať na danom serveri.

## 6.2. Manažovateľný prepínač

Testovanie prebehlo vytvorením a spustením viacerých simulácií obsahujúcich manažovateľný prepínač. Priebeh testovania bol bezproblémový, nami implementované riešenie funguje bez akýchkoľvek ťažkostí. Stále však pretrvávajú spomenuté nevýhody, ktoré znemožňujú použitie iného STP protokolu ako 802.1d a neumožňujú nastavovať zabezpečenia konkrétnych portov.

## 6.3. PIX Firewall

Testovanie prebiehalo spúšťaním simulácií s rôznym počtom PIX *firewall-ov*. Na jednom z počítačov, na ktorom bol projekt vyvíjaný, sa objavil problém s nekompatibilitou programu Pemu s HW architektúrou počítača. Po pripojení sa na konzolu konkrétnej inštancie *firewall-u* PIX zostala na termináli čierna obrazovka a nič nebolo možné konfigurovať. Navonok nehlásil Pemu server žiadne chyby, preto sme na tento problém prišli až pri testovaní. Zistili sme tiež, že na tejto architektúre dokonca nie je možné skompilovať zdrojové súbory. Preto bolo tento problém nutné vyriešiť migráciou projektu na iný hardvér, ktorý s programom Pemu už spolupracovať vedel.

Počas testovania sa po odladení skriptov ďalšie problémy nepreukázali. Bohužiaľ PIX zaťažuje aj po použití programu *cpulimit* procesor značne viac ako smerovač v programe Dynagen. Simulácie obsahujúce PIX *firewall-y* sú teda oveľa náročnejšie na výpočtový výkon a budú pravdepodobne v praxi používané zriedkavejšie a v menšom počte.

## 6.4. Laboratórne cvičenia

V prvom kroku sme testovali či je možné pripojiť sa na všetky zariadenia, ktoré sú v jednotlivých simuláciách. V prípade, že pripojenie nebolo možné sme problém našli a odstránili. V súčasnom stave sú teda pripojenia na zariadenia aktuálnej simulácie plne funkčné.

Ďalším krokom bola kontrola topológií a konfigurácií, či:

- sú zariadenia spájané podľa zadania,
- bol obrázok topológie vygenerovaný správne,
- je obrázok topológie aktuálny (po zmene konfigurácie ho treba pregenerovať),
- či v konfigurácii nie sú nejaké chyby a pod.

Posledným krokom testovania laboratórnych cvičení, bolo zisťovanie záťaže simulácií na server. Najprv sme testovali simulácie cvičení jednotlivo a v prípade potreby sme upravili potrebné hodnoty využitia pamäte a procesora pre zariadenia. Po odladení sme skúšali simuláciu viacerých cvičení naraz a testovali sme tak aj maximálnu záťaž, ktorú vydrží aktuálny server a simulácie pritom bežia bez problémov. Podľa týchto testov sme nastavili maximálne zaťaženie CPU servera na 85 % a maximálne zaťaženie pamäte servera na 4 GB.

## 7. Záver

Hlavným cieľom nášho tímového projektu bolo vytvoriť simulátor komunikácie v počítačovej sieti. Po dohode s vedúcim projektu sme sa rozhodli pokračovať v projekte z minulého roka a doviesť ho do takého stavu, aby sa dal použiť ako učebná pomôcka pri výučbe predmetov ako sú Počítačové siete, WAN technológie a podobne.

V spolupráci s Bc. Petrom Pétiom, ktorý mal na starosti webové rozhranie projektu ako diplomovú prácu, sa nám podarilo vytvoriť funkčné virtuálne laboratórium, ktoré spĺňa stanovené požiadavky. Jeho primárnou funkciou je možnosť simulácie pripravených laboratórnych cvičení rôznych úrovní. Používateľ tak má možnosť vyskúšať si nielen základné konfigurácie smerovačov a prepínačov, ale aj zložitejšie topológie či dokonca konfigurácie PIX *firewall-ov*. Veľkou výhodou je neustála dostupnosť virtuálneho laboratória zo siete Internet oproti obmedzenému prístupu ku klasickému laboratóriu v škole. Študent si tak môže doma vyskúšať to, čo v škole nestihol, prípadne to čo ho zaujíma. Pre vyučujúceho je to jednoduchý nástroj na získanie informácií o tom, ako často sa študenti venujú problematike daného predmetu. Vyučujúci môže kedykoľvek upraviť, prípadne doplniť laboratórne cvičenia, čo prináša pre obe strany dostatočnú flexibilitu v možnostiach využitia virtuálneho laboratória.

Momentálne najväčšou nevýhodou celého projektu je server, ktorý nemá dostatočne veľké zdroje. V súčasnosti môžu naraz simulovať maximálne štyria používatelia naraz. Nie je to síce veľa, ale keď si uvedomíme, že virtuálne laboratórium je prístupné 24 hodín 7 dní v týždni, nie je to až tak málo. V prípade využívania nášho projektu na cvičeniach predmetov Počítačové siete 2 a WAN technológie odporúčame zvýšiť pamäť a procesorový výkon aktuálneho serveru, alebo premiestniť projekt na výkonnejší server. Ďalšou možnosťou je využiť „*GRID computing*“ a s jeho pomocou dosiahnuť omnoho väčší výpočtový výkon a teda umožniť tak viacerým študentom naraz využívať virtuálne laboratórium.

Ďalším zámerom tímového projektu bolo vyskúšať si spoluprácu v tíme. Väčšina z nás už tieto skúsenosti mala a pri práci na projekte sme sa navzájom dopĺňali. Každý zastával svoju pozíciu a v prípade potreby komunikoval či spolupracoval s ostatnými bez väčších problémov.



## 8. Zoznam literatúry

- [1] HAMŠÍK, A., SCHMOTZER, M., MÓZES, T., PÉTI, P., NEMSILA, M.: Simulátor komunikácie v počítačovej sieti. 2008. 85 s. Slovenská technická univerzita v Bratislave, Fakulta informatiky a informačných technológií. Vedúci Tímového projektu Ing. Grellneth Igor, PhD. Dostupný z WWW: <[http://labss2.fiit.stuba.sk/TeamProject/2007/team01pss/subory/lis\\_dok.pdf](http://labss2.fiit.stuba.sk/TeamProject/2007/team01pss/subory/lis_dok.pdf)>.
- [2] Request for Comments [online]. 1992- , 28Jul04 [cit. 2008-30-10]. Dostupný z WWW: <<http://www.rfc-editor.org/rfc.html>>.
- [3] W. CHAMPAN JR., David, FOX, Andy. Zabezpečení sítí pomocí Cisco PIX Firewall. Preložil David Krásenský. Brno : Computer Press, 2004. 343 s.
- [4] HAMŠÍK, A., SCHMOTZER, M., MÓZES, T., PÉTI, P., NEMSILA, M.: Simulátor komunikácie v počítačovej sieti - Používateľská príručka. 2008 . 9 s. Slovenská technická univerzita v Bratislave, Fakulta informatiky a informačných technológií. Vedúci Tímového projektu Ing. Grellneth Igor, PhD. Dostupný z WWW:  
<<http://labss2.fiit.stuba.sk/TeamProject/2007/team01pss/subory/prirucka.pdf>>

## 9. Prílohy

V tejto časti nájdete všetky prílohy spomenuté v dokumentácii ako napríklad používateľská príručka alebo inštalačná príručka.

# **A. Riadenie projektu**

## **B. Posudky**

## C. Používateľská príručka – VLAB

V tejto príručke nájdete vysvetlenia a návody ako používať jednotlivé časti systému. Každý z používateľov má rôzne privilégia. Študent má najmenej práv a všetky sú popísané v prvej časti tejto príručky. Učiteľ môže využívať všetky funkcionality, ktoré má k dispozícii študent (prvá časť príručky). Funkcionality, ktoré študent už nemá sú uvedené v druhej časti príručky. V tretej časti používateľskej príručky sú popísané možnosti použitia systému pre administrátora. Administrátor samozrejme môže používať to čo učiteľ a študent.

Finálna verzia systému VLAB je sprístupnená na adrese <https://vlab.fiit.stuba.sk>. Používateľské rozhranie je optimalizované pre prehliadač Mozilla FireFox, v iných prehliadačoch sa môžu vyskytnúť odchýlky v zobrazení jednotlivých prvkov.

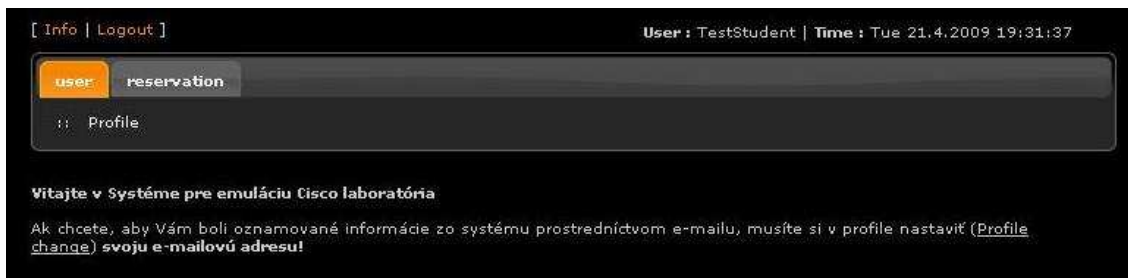
### A.1. Študent

#### A.1.1. Prihlásenie

Pre prístup do systému používateľ získa od svojho cvičiaceho prihlasovacie meno a heslo. Pomocou týchto údajov sa prihlási na úvodnej prihlasovacej obrazovke:

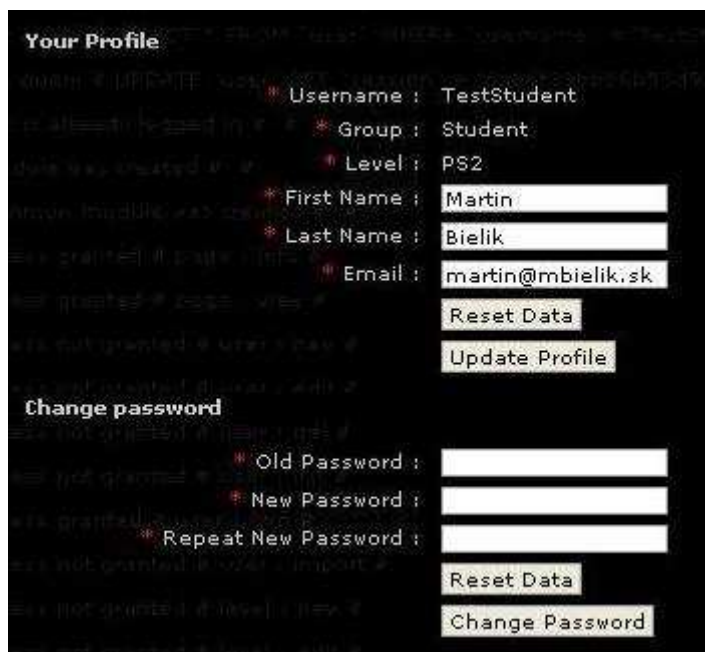


Po prihlásení sa zobrazí základná obrazovka používateľa zobrazujúca meno prihláseného používateľa, hlavné menu systému prístupné pre daného používateľa a aktuálny čas. Sem sa môže používateľ kedykoľvek vrátiť kliknutím na tlačidlo **INFO**:



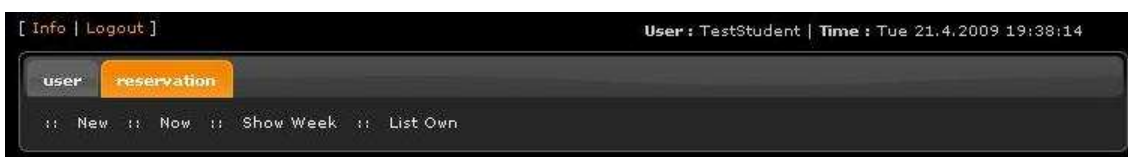
## A.1.2. Menu USER

Po kliknutí na položku **USER** sa zobrazí menu používateľa. Po kliknutí na **PROFILE** sa zobrazí profil prihláseného používateľa, v ktorom môže upraviť svoje meno, priezvisko, e-mail a zmeniť si heslo:



## A.1.3. Menu RESERVATION

Po kliknutí na **RESERVATION** sa sprístupní menu rezervácií, ktoré obsahuje položky **NEW**, **NOW**, **SHOW WEEK** a **LIST OWN**:



## Simulátor komunikácie v počítačovej sieti

Po kliknutí na **NEW** sa zobrazí sprievodca pridaním novej rezervácie. V prvom kroku treba vybrať deň, v ktorom si chce študent vytvoriť rezerváciu:



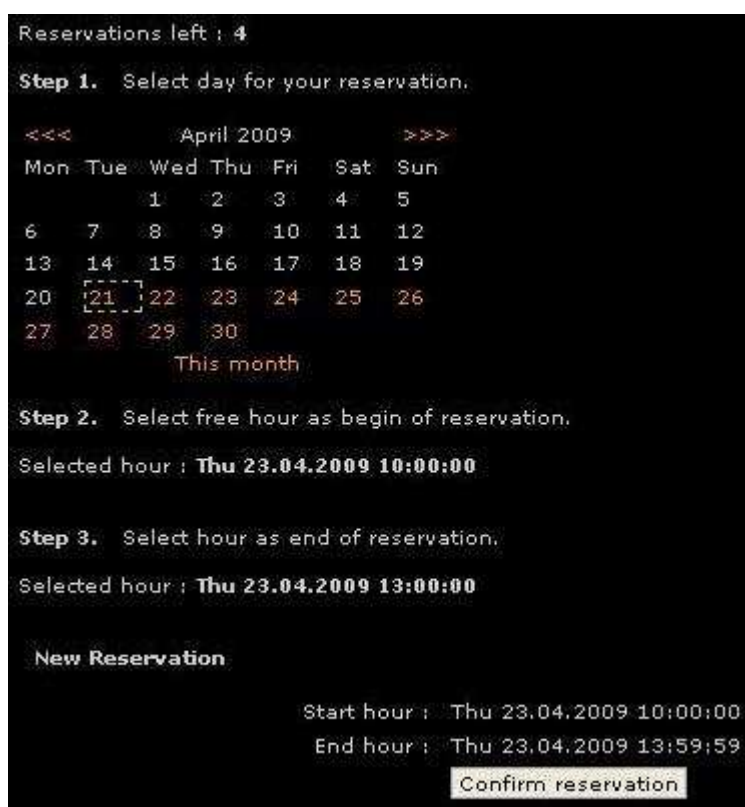
Po kliknutí na príslušný deň (oranžovou farbou sú zobrazené prístupné dni) sa zobrazí možnosť výberu začiatkovej hodiny rezervácie:



Po kliknutí na príslušnú hodinu (prístupné hodiny sú zobrazené oranžovým písmom a čiernym podfarbením) sa zobrazí možnosť výberu času ukončenia rezervácie (možné trvanie rezervácie, ako aj maximálny počet rezervácií závisí na tom, do akej úrovne používateľ patrí):

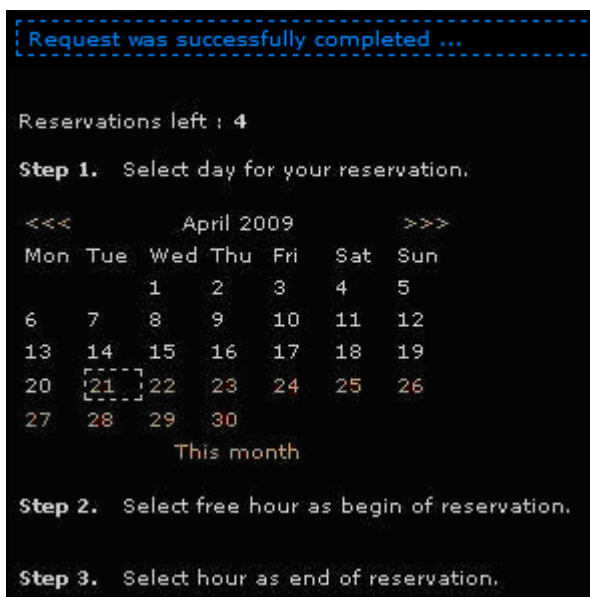


Po kliknutí na príslušnú hodinu ukončenia rezervácie sa zobrazí zhrnutie rezervácie, ktoré si študent skontroluje a potvrdí tlačidlom:

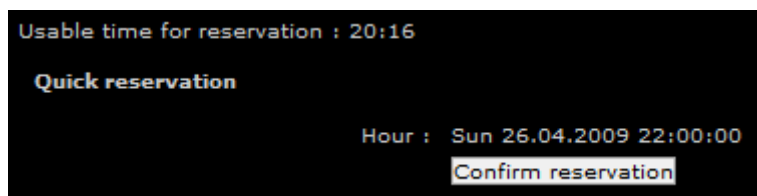


Po kliknutí na **CONFIRM RESERVATION** sa zobrazí hlásenie o úspešnom pridaní rezervácie a študent si môže rezervovať ďalšie časové okno:

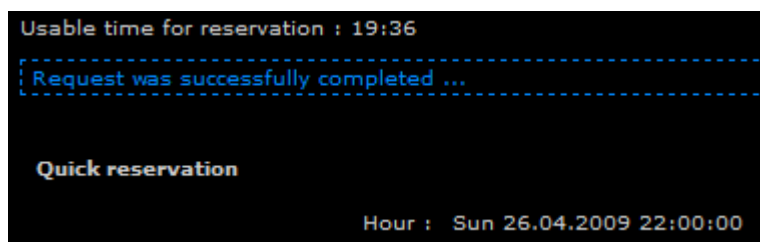




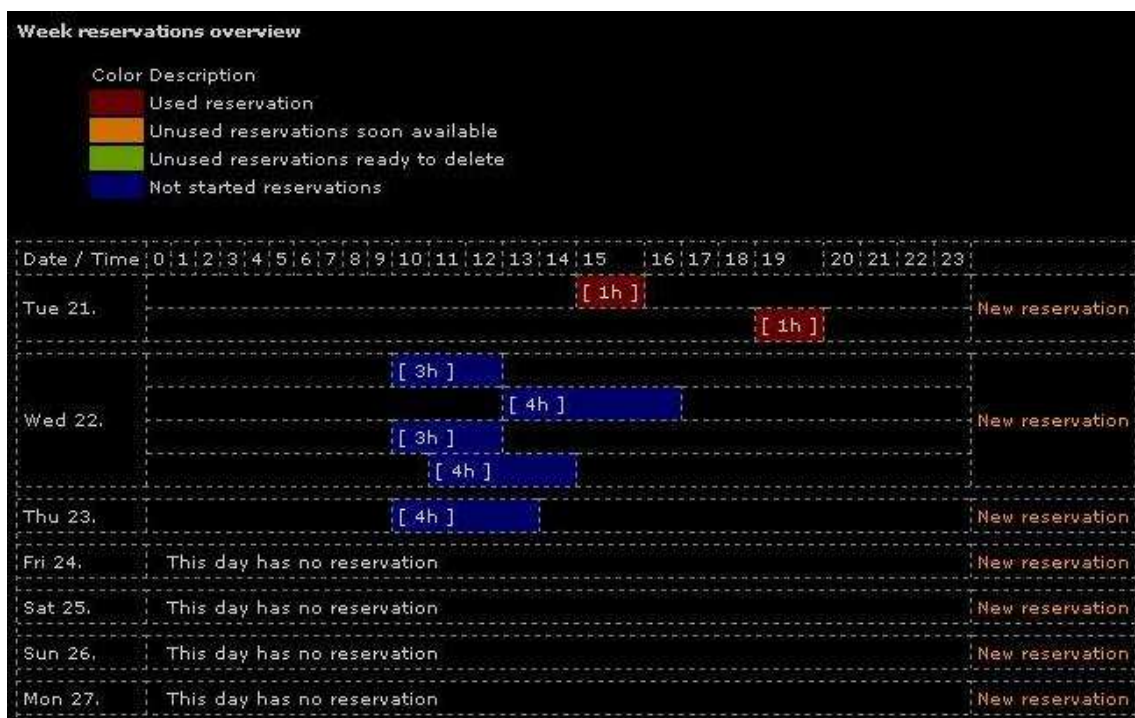
Po kliknutí na **NOW** v hlavnom menu sa zobrazí možnosť vytvoriť rezerváciu na aktuálnu hodinu, čo znamená od teraz po koniec práve prebiehajúcej hodiny. Takáto rezervácia je možná len v prípade, že systém má dostatok voľných prostriedkov pre vytvorenie rezervácie používateľa aktuálnej úrovne. Je tu aj zobrazený čas, koľko zostáva do konca celej hodiny, takže študent má prehľad, koľko času touto rezerváciou získa. Takúto rezerváciu si môže používateľ vytvoriť minimálne 15 minút pred celou hodinou:



Po kliknutí na **CONFIRM RESERVATION** sa zobrazí hlásenie o úspešnom pridaní rezervácie:



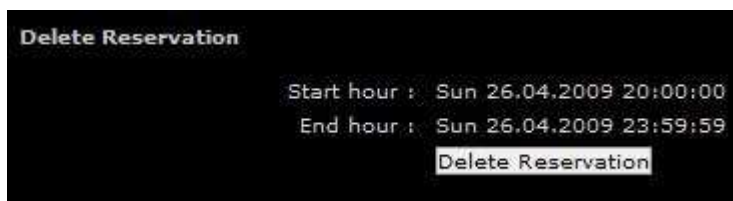
Po kliknutí na **SHOW WEEK** sa zobrazí aktuálny týždeň s rezerváciami všetkých používateľov. Rezervácie sú farebne odlišené, pričom modrá farba znamená ešte nespustenú rezerváciu. Oranžová znamená, že čas danej rezervácie už nastal, ale daný študent ju zatiaľ nevyužil. Zelená farba znamená, že rezervácia nebola využitá ani 30 minút po jej začiatku a preto je sprístupnená pre iných študentov, ktorí ju môžu zmazať. Červená farba znamená využitú rezerváciu:



Po kliknutí na **LIST OWN** sa študentovi zobrazia všetky jeho rezervácie s informáciami o čase štartu a ukončenia rezervácie, o tom, či je daná rezervácia aktívna a koľko simulácií prebehlo v rámci danej rezervácii:

Start hour	End hour	Active	Sim. count	
Mon 27.04.2009 00:00:00	Mon 27.04.2009 03:59:59	No	0	Simulations   Delete
Sun 26.04.2009 20:00:00	Sun 26.04.2009 23:59:59	Yes	0	Simulations   Free resources
Sat 25.04.2009 16:00:00	Sat 25.04.2009 16:59:59	No	0	Simulations
Sat 25.04.2009 15:00:00	Sat 25.04.2009 15:59:59	No	0	Simulations
Wed 22.04.2009 22:00:00	Wed 22.04.2009 23:59:59	No	0	Simulations
Wed 22.04.2009 21:00:00	Wed 22.04.2009 21:59:59	No	0	Simulations
Wed 22.04.2009 12:00:00	Wed 22.04.2009 12:59:59	No	1	Simulations
Tue 21.04.2009 21:00:00	Tue 21.04.2009 21:59:59	No	0	Simulations
Fri 17.04.2009 10:00:00	Fri 17.04.2009 19:59:59	No	0	Simulations
Wed 15.04.2009 12:00:00	Wed 15.04.2009 14:59:59	No	1	Simulations
Thu 09.04.2009 06:00:00	Thu 09.04.2009 07:59:59	No	0	Simulations
Tue 07.04.2009 13:00:00	Tue 07.04.2009 17:59:59	No	6	Simulations
Sun 05.04.2009 22:00:00	Sun 05.04.2009 23:59:59	No	2	Simulations

V prípade, že simulácia ešte nenastala, používateľ má možnosť takúto rezerváciu zmazať tlačidlom **DELETE** v príslušnom riadku rezervácie. Pre prípad, že v danom čase rezerváciu nevyužije a uvoľní tak priestor pre ďalšie rezervácie. V prípade, že simulácia už začala, ale používateľ skončil skorej, má taktiež možnosť uvoľniť systémové prostriedky serveru pre ďalšie využitie kliknutím na **FREE RESOURCES**. V tomto prípade sa zmaže rezervácia od najbližšej celej nezačatej hodiny až do konca rezervácie:



Po potvrdení rezervácie dostaneme hlásenie o úspešnom vykonaní zvolenej akcie:

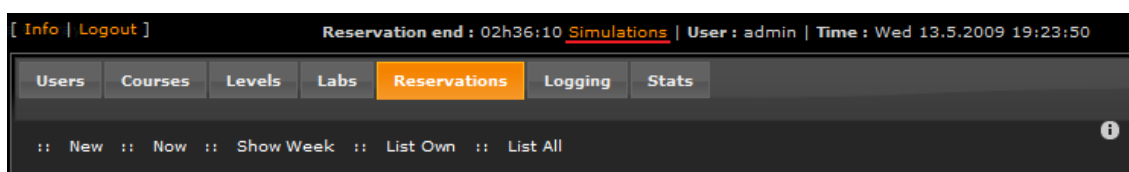


Po kliknutí na **SIMULATIONS** v riadku príslušnej rezervácie sa zobrazí zoznam simulácií, ktoré boli spustené v rámci danej rezervácie a je tu aj možnosť prezrieť si topológiu simulovaného laboratórneho cvičenia kliknutím na **SHOW PICTURE**:

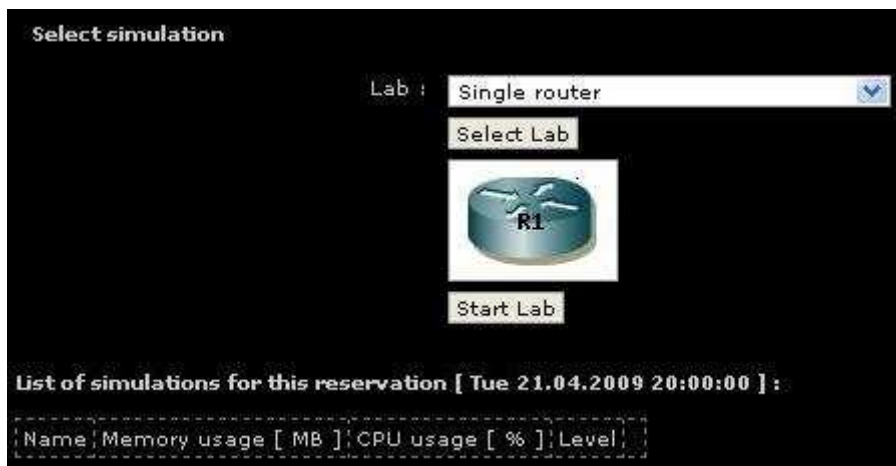
List of simulations for this reservation [ Tue 07.04.2009 00:00:00 ] :

Name	Memory usage [ MB ]	CPU usage [ % ]	Level	
Základná konfigurácia smerovača	160	7	PS2	Show picture
Základná konfigurácia smerovača	160	7	PS2	Show picture
Základná konfigurácia smerovača	160	7	PS2	Show picture

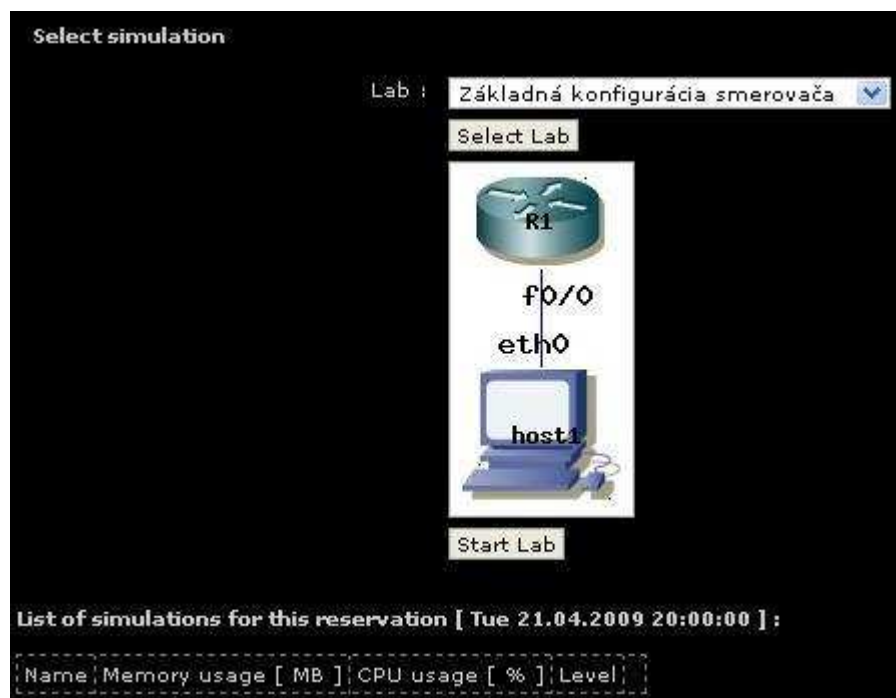
Ak je daná rezervácia v stave **ACTIVE**, znamená to, že je aktívna a je možné v nej vykonať simuláciu. Pre zjednodušenie stačí kliknúť na **SIMULATIONS** v pravom hornom rohu. Táto možnosť sa zobrazí len v čase, keď má používateľ aktívnu rezerváciu:



Po kliknutí na **SIMULATIONS** sa zobrazí možnosť výberu simulácie. Každý študent si môže vyberať simulácie s rôznou zložitosťou podľa úrovne, do ktorej je zaradený:



Po zvolení príslušnej simulácie a kliknutí na **SELECT LAB** sa zobrazí obrázok topológie danej simulácie, ktorá je pripravená na štart:



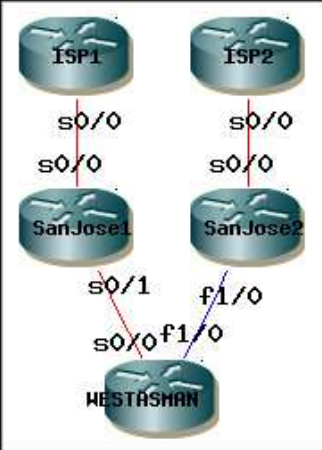
Po kliknutí na **START LAB** v spodnej tabuľke pribudne práve naštartovaná simulácia a tlačidlá na spustenie konzoly pre jednotlivé zariadenia. V časti **ASSIGNMENT** študent nájde krátky popis k zobrazenej topológii a odkaz na súbor s pripraveným laboratórnym cvičením pre danú topológiu:

## Simulátor komunikácie v počítačovej sieti

**Connect to simulation**

Assignment :

Základnú konfiguráciu BGP si môžete odskúšať podľa laboratórneho cvičenia, ktoré nájdete na [http://student.fiit.stuba.sk/~schuster05/TP/WAN/WAN\\_BasicBGP.pdf](http://student.fiit.stuba.sk/~schuster05/TP/WAN/WAN_BasicBGP.pdf).



Device : **WESTASMAN**

**Connect**

**Connect and detach console**

**Connect SSH**

Warning : Save your running config ( router# copy run start ) before simulation stop otherwise it will be not saved into database and available for later use.

**Stop active simulation**

Po kliknutí na **CONNECT SSH** sa zobrazí adresa, používateľské meno a heslo pre pripojenie na zvolené zariadenie. Pomocou týchto údajov sa môže používateľ napríklad cez program *putty* prihlásiť na jednotlivé konzoly aktívnych zariadení prebiehajúcej simulácie.

**Connect to simulation**



Device : **R1**

**Connect**

**Connect and detach console**

**Connect SSH**

Warning : Save your running config ( router# copy run start ) before simulation stop otherwise it will be not saved into database and available for later use.

**Stop active simulation**

**SSH connection :**

**server :** vlab.fiit.stuba.sk  
**port :** 22  
**username :** teststudent\_7503  
**password :** heslo



Tlačidlo **STOP ACTIVE SIMULATION** zastaví aktuálne prebiehajúcu simuláciu. Používa sa v prípade, že študent už prácu ukončil a nechystá sa pokračovať. Po ukončení simulácie si môže používateľ pozrieť zoznam simulácií, v ktorom okrem iného nájde obrázok použitej topológie či konfigurácie jednotlivých zariadení, ale to len v prípade, že si konfiguráciu pred ukončením simulácie uložil do NVRAM. Po kliknutí na **SHOW CONFIG** v príslušnom riadku simulácie sa zobrazia zariadenia s jednotlivými konfiguráciami (ak konfiguráciu na zariadení používateľ neuložil, bude súbor prázdny):

List of all made simulations for this reservation [ Mon 11.05.2009 17:00:00 ] : 147.175.182.24', 'last' =>

Name	Memory usage [ MB ]	CPU usage [ % ]	Level	Active	
Pokročilé smerovanie	480	21	CCNA	No	Show picture   Show config
Router-Host	320	14	WAN	No	Show picture   Show config
Basic BGP	640	25	WAN	No	Show picture   Show config
Router-Host	320	14	WAN	No	Show picture   Show config
Router-Host	320	14	WAN	No	Show picture   Show config
Router-Host	320	14	WAN	No	Show picture   Show config

R1BA    **R1PRAHA**

```

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname mmmmmm
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
!

```

## A.2. Učiteľ

Učiteľ môže vykonávať tie isté činnosti ako študent a má ďalšie nasledovné možnosti konfigurácie systému. Po prihlásení sa mu zobrazí základné menu:

[ Info | Logout ]      User : TestTeacher | Time : Fri 24.4.2009 22:19:59

user   course   level   lab   reservation

:: New   :: List   :: Profile

Vitajte v Systéme pre emuláciu Cisco laboratória

Ak chcete, aby Vám boli oznamované informácie zo systému prostredníctvom e-mailu, musíte si v profile nastaviť (Profile change) svoju e-mailovú adresu!

## A.2.1. Menu USER

V menu **USER** sa nachádza položka **NEW**, ktorá dovoľuje učiteľovi pridať nového používateľa systému. Je potrebné vyplniť prihlasovacie meno, heslo, skupinu, úroveň, kurz (trieda), meno, priezvisko a e-mail nového používateľa:

**New User**

Username :

Password :

Repeat New Password :

Group : Teacher ▾

Level : PS2 ▾

Course : - ▾

First Name :

Last Name :

Email :

Reset Data

Add New User

Po kliknutí na **LIST** sa učiteľovi zobrazí zoznam používateľov systému, ktorý môže filtrovať pomocou viacerých položiek, napríklad meno, alebo skupina. V poslednom stĺpci má možnosť upraviť alebo zmazať profil študenta, nie však administrátora:

**Filter Users**

Username :

Group : Doesn't matter ▾

Level : Doesn't matter ▾

Course : Doesn't matter ▾

First Name :

Last Name :

Email :

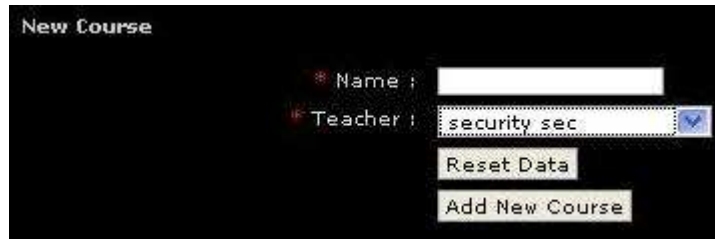
Reset Data

Filter Users

Username	Group	Level	Course	First name	Last name	Email	
pit	Administrator	WAN	-		PIT	pit@pit.sk	
peter	Student	WAN	WAN_09	peter	havrila	phavrila@gmail.com	Edit   Delete
filip	Student	WAN	WAN_09	filip	burda	filip.burda@gmail.com	Edit   Delete
security	Teacher	WAN	WAN_09	security	sec	macko@ynet.sk	Edit   Delete
ucitel8	Teacher	WAN	WAN_09	ucitel8	team8	team8@team8.sk	Edit   Delete
mato	Administrator	CCNA	-	Martin	Bielik	martin@mbielik.sk	
lowli	Administrator	SEC	-	Peter	Marko	lowli86@gmail.com	
admin	Administrator	SEC	-	admin	Admin	admin@vlab.net	
igor	Student	SEC	SEC_09	igor	igor	grellneth@fiit.stuba.sk	Edit   Delete
11111	Student	SEC	SEC_09	meno	priezvisko	xpriezvisko@is.stuba.sk	Edit   Delete
57893	Student	SEC	WAN_09	jfwoia	jfwoie	xjfwoie@is.stuba.sk	Edit   Delete
3940	Student	SEC	SEC_09	kfpwi	fwo	xjwo@is.stuba.sk	Edit   Delete
392	Student	SEC	SEC_09	fkwoa	kfwp	xkfwp@is.stuba.sk	Edit   Delete

## A.2.2. Menu COURSE

V menu **COURSE** sa nachádza položka **NEW**, ktorá umožňuje pridať nový kurz či triedu do systému. Je potrebné vyplniť meno nového kurzu a prideliť mu učiteľa:



New Course

Name :

Teacher : security sec

Reset Data

Add New Course

Kliknutím na **LIST** sa zobrazí zoznam kurzov vytvorených v systéme, ktoré je možné takisto filtrovať podľa jeho názvu alebo učiteľa:



Filter Course

Name :

Teacher : Doesn't matter

Reset Data

Filter Course

Name	Teacher	
WAN_09	zuzana schusterova	Edit   Delete
PS2_08	Martin Bielik	Edit   Delete
SEC_09	security sec	Edit   Delete
CCNA	ucitel8 team8	Edit   Delete

## A.2.3. Menu LEVEL

V menu **LEVEL** sa nachádza položka **NEW**, ktorá umožňuje pridanie novej úrovne simulácie. Na jej vytvorenie je potrebné zadať jeho meno, maximálny počet rezervácií prípustných pre danú úroveň, maximálnu možnú využiteľnú pamäť, ako aj maximálne zaťaženie procesora jednou simuláciou používateľa z tejto úrovne a maximálne prípustné okno rezervácie (čo vyjadruje, koľko najviac hodín si môže používateľ z tejto úrovne naraz rezervovať):



Po kliknutí na **LIST** sa zobrazí zoznam úrovní vytvorených v systéme. Dôležité je, že vyššia úroveň obsahuje nielen svoje laboratórne cvičenia, ale aj cvičenia predchádzajúcich úrovní. V našom prípade najnižšia úroveň sa nachádza navrchu tabuľky a najvyššia úroveň zasa na konci tabuľky. Poradie môžeme zmeniť pomocou tlačidiel **UP** a **DOWN** v príslušnom riadku úrovne:

Name	Max. reservation count	Max. memory usage [ MB ]	Max. cpu usage [ % ]	Max. time window	
PS2	4	512	21	4	Edit   Delete   Up   Down
WAN	2	800	35	6	Edit   Delete   Up   Down
CCNA	2	1024	35	8	Edit   Delete   Up   Down
SEC	2	1024	40	10	Edit   Delete   Up   Down

## A.2.4. Menu LAB

V menu **LAB** sa nachádza položka **NEW**, ktorá umožňuje pridať nové laboratórne cvičenie do systému. Pri pridávaní je nutné zadať meno, úroveň, do ktorého cvičenie patrí a konfiguráciu pre program Dynagen. Po zadaní týchto parametrov je potrebné vygenerovať obrázok topológie a to pomocou tlačidla **RELOAD TOPOLOGY PICTURE**. Ak sme zadali správnu konfiguráciu, vygeneruje sa nám obrázok, podľa ktorého vidíme akými káblami sú jednotlivé zariadenia spojené medzi sebou a hlavne na ktorých portoch. V prípade, že sa obrázok nevygeneruje, alebo nie je podľa predstáv, pravdepodobne je chyba v konfigurácii, treba ju opraviť a následne znovu vygenerovať obrázok. Ak na tento krok zabudneme, môže sa stať, že obrázok nebude zodpovedať skutočnej topológii a bude nepoužiteľný. Spolu s vygenerovaním obrázku topológie sa automaticky prepočítavajú aj hodnoty **MEMORY USAGE** a **CPU USAGE**. Môže sa potom stať, že úroveň do ktorej sme toto cvičenie zaradili nebude postačujúca a budeme ju musieť zmeniť. Ak máme všetko správne vyplnené, pre pridanie nového laboratórneho cvičenia klikneme na **ADD NEW LAB** a zobrazí sa nám informácia o jeho úspešnom pridaní do zoznamu cvičení. Kliknutím na **RESET DATA** zmažeme všetko zadané a môžeme začať odznova.



Po kliknutí na **LIST** sa zobrazí zoznam laboratórnych cvičení pridaných do systému, v ktorom je možné filtrovať napríklad podľa mena labu, podľa autora a pod.:

**Filter Labs**

Name :

Memory usage [ MB ] :

CPU usage [ % ] :

Configuration :

Author : Doesn't matter ▾

Level : Doesn't matter ▾

Name	Memory usage [ MB ]	CPU usage [ % ]	Author	Level	
BGP	640	25	admin Admin	WAN	Show picture   Show config   Edit   Delete
Frame Relay 1	512	20	admin Admin	WAN	Show picture   Show config   Edit   Delete
Frame Relay 2	512	20	admin Admin	WAN	Show picture   Show config   Edit   Delete

## A.2.5. Menu RESERVATION

Učiteľ má navyše od študenta možnosť prezrieť si rezervácie všetkých používateľov v systéme kliknutím na **LIST ALL**. Táto časť taktiež obsahuje filter, na základe ktorého si môže používateľ vybrať, čo sa mu má zobraziť:

Filter Reservations

Username :

Group : Doesn't matter ▾

First Name :

Last Name :

Email :

Date From : 4/23/2009

Date To : 4/30/2009

Reset Data

Filter Users

Click on reservation for details.

Date / Time	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
Thu 23, 04, 2009													security sec [ 1h ]	security sec [ 1h ]	security sec [ 1h ]									admin Admin [ 2h ]	
Fri 24, 04, 2009																									admin Admin [ 10h ]

Po kliknutí na rezerváciu sa zobrazia jej detaily:

Reservation details

Date : Sat 25. 04. 2009

User : peter havrila [ WAN ]

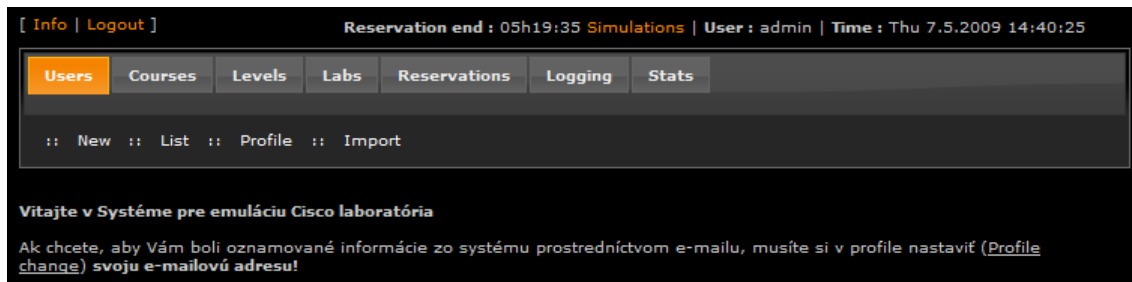
Time : 21:00 - 22:59 [ 2h ]

Show simulations Delete reservation Close

Tu máme možnosť pozrieť si študentove simulácie, ak nejaké už prebehli a to stlačením **SHOW SIMULATIONS**. V prípade, že simulácia ešte neprebehla, môžeme ju zrušiť ak je to potrebné využitím tlačidla **DELETE RESERVATION**.

## A.3. Administrátor

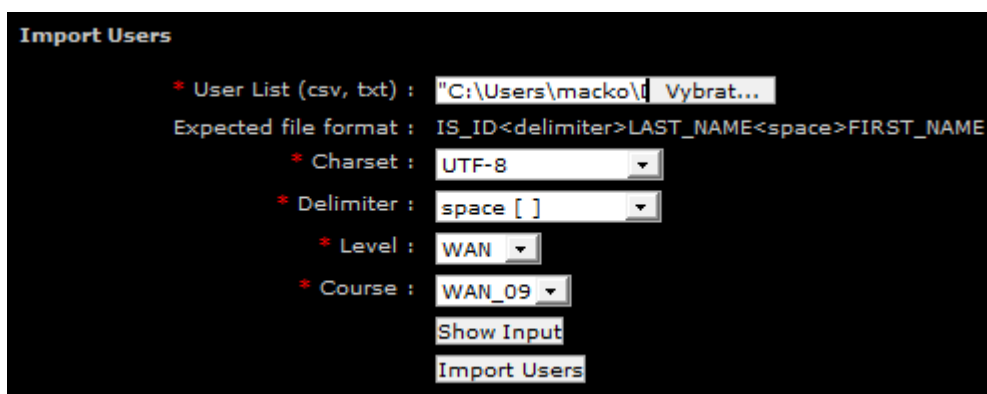
Administrátor môže vykonávať tie isté činnosti ako učiteľ a navyše má ďalšie možnosti konfigurácie systému. Po prihlásení sa učiteľa sa mu zobrazí základné menu:



### A.3.1. Menu USER

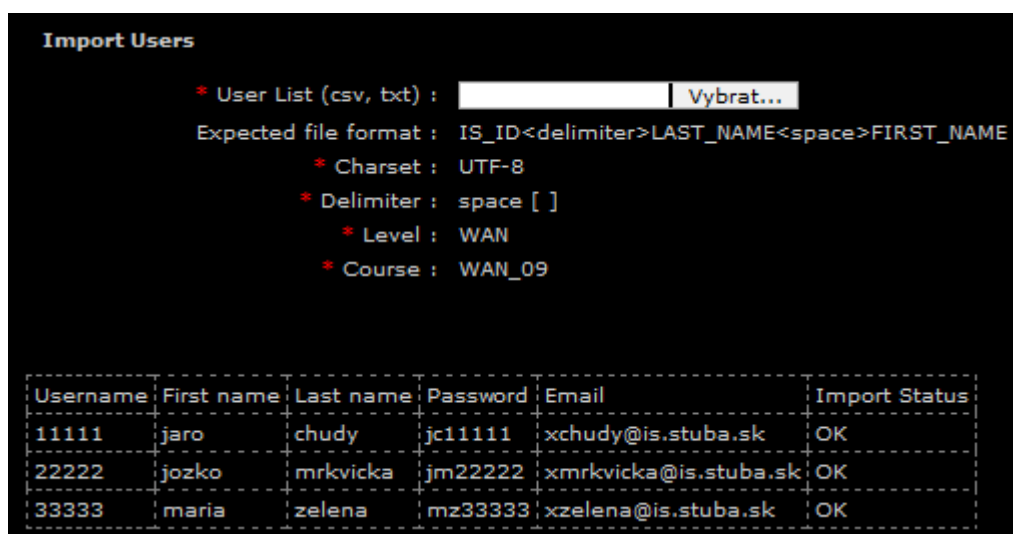
Administrátor má navyše od učiteľa možnosť editovať a mazať všetkých používateľov v systéme. Existuje jeden hlavný administrátor, ktorého nemožno odstrániť (pre prípad, že by sa niekto rozhodol zmazať všetkých používateľov), aby sa dalo do systému vždy prihlásiť.

Ďalšou možnosťou administrátora je importovať používateľov systému zo súboru. Pri tejto akcii je nutné vybrať požadovaný súbor so študentmi pomocou tlačidla v danom riadku, následne zadať znakovú sadu, oddeľovač jednotlivých stĺpcov v súbore, úroveň, do ktorej sa majú študenti importovať a prípadne kurz. Importovaný súbor si môžeme najprv prezrieť stlačením ***SHOW INPUT*** alebo hneď vložiť pomocou tlačidla ***IMPORT USER***.



V prípade chybného súboru systém vypíše chybové hlásenie, v opačnom prípade vypíše oznam o úspešnom importovaní používateľov do systému a zobrazí

vložených používateľov, spolu s prednastavenými heslami (v tvare iniciály a ID z akademického informačného systému - AIS) a emailami (predpokladajú sa vstupné údaje z AIS):



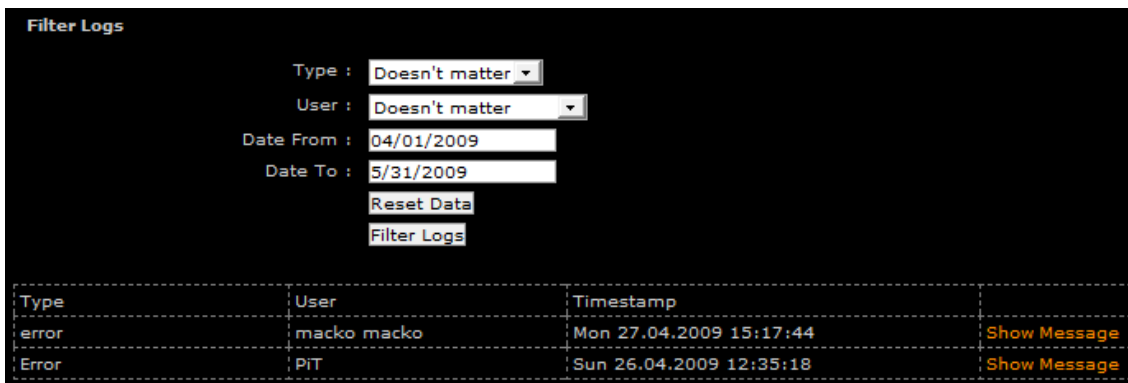
Tu vidíte ukážku použitého vstupného súboru *import.txt* pre importovanie používateľov:

```
11111 chudy jaro
22222 mrkvicka jozko
33333 zelena maria
```

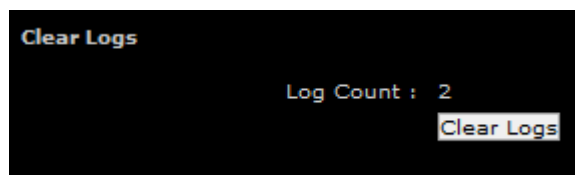
V prvom stĺpci sa nachádzajú identifikačné čísla študentov, potom priezvisko a meno oddelené medzerou.

### A.3.2. Menu LOGGING

V menu **LOGGING** v položke **LIST** môžete nájsť všetky záznamy chybových výpisov systému. Záznamy sa zobrazujú len z určitého časového intervalu, ktorý si môžeme jednoducho zmeniť a v prípade potreby filtrovať dáta ešte podľa typu či používateľa. Po kliknutí na **FILTER LOGS** sa nám zobrazia požadované záznamy. Pre zobrazenie podrobného výpisu kliknite na položku **SHOW MESSAGE**:

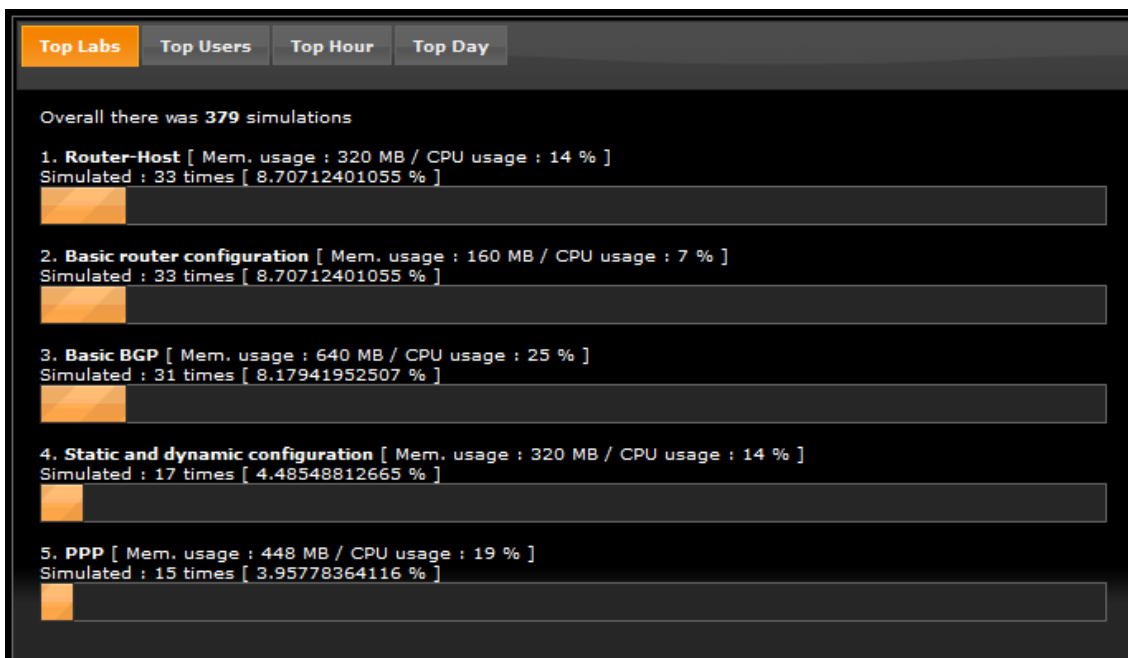


Ďalšou možnosťou v tomto menu je vymazanie záznamov v časti **CLEAR LOGS**:



### A.3.3. Menu STATS

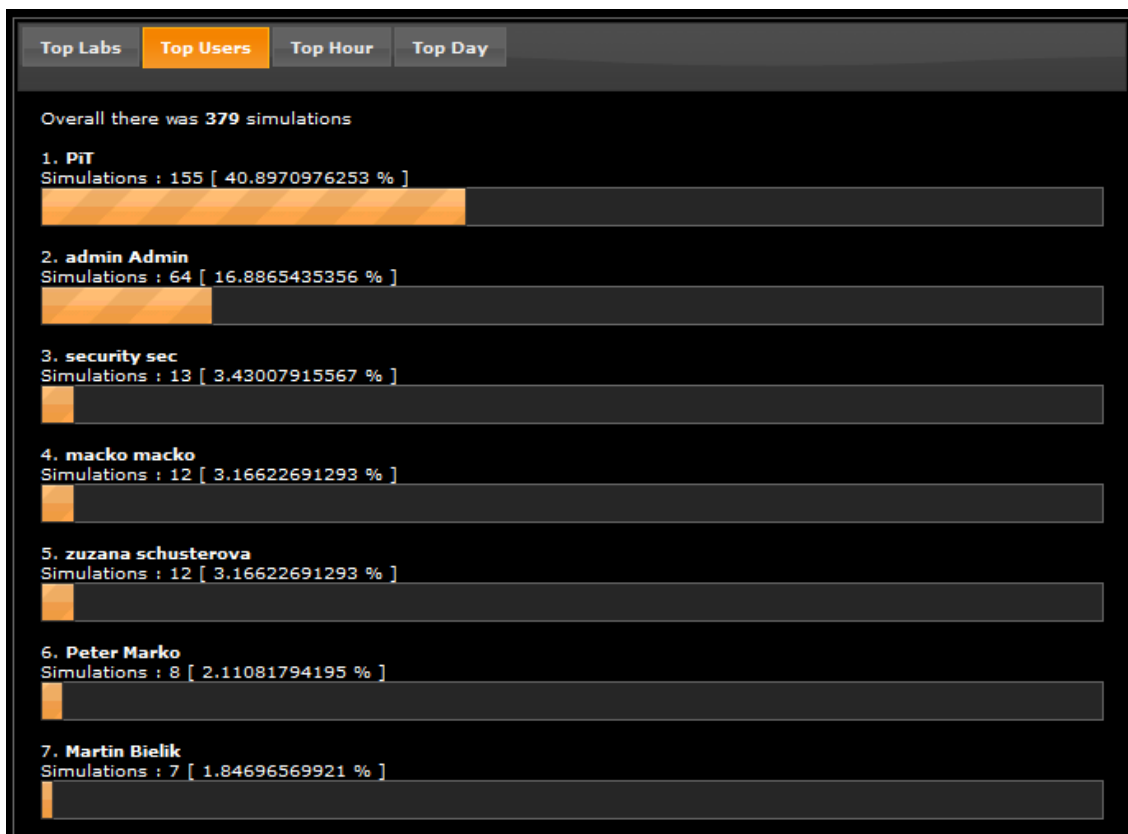
V tejto časti si administrátor môže pozrieť štatistiky využívania systému. Po kliknutí na položku **SHOW** sa zobrazí nasledovná tabuľka:



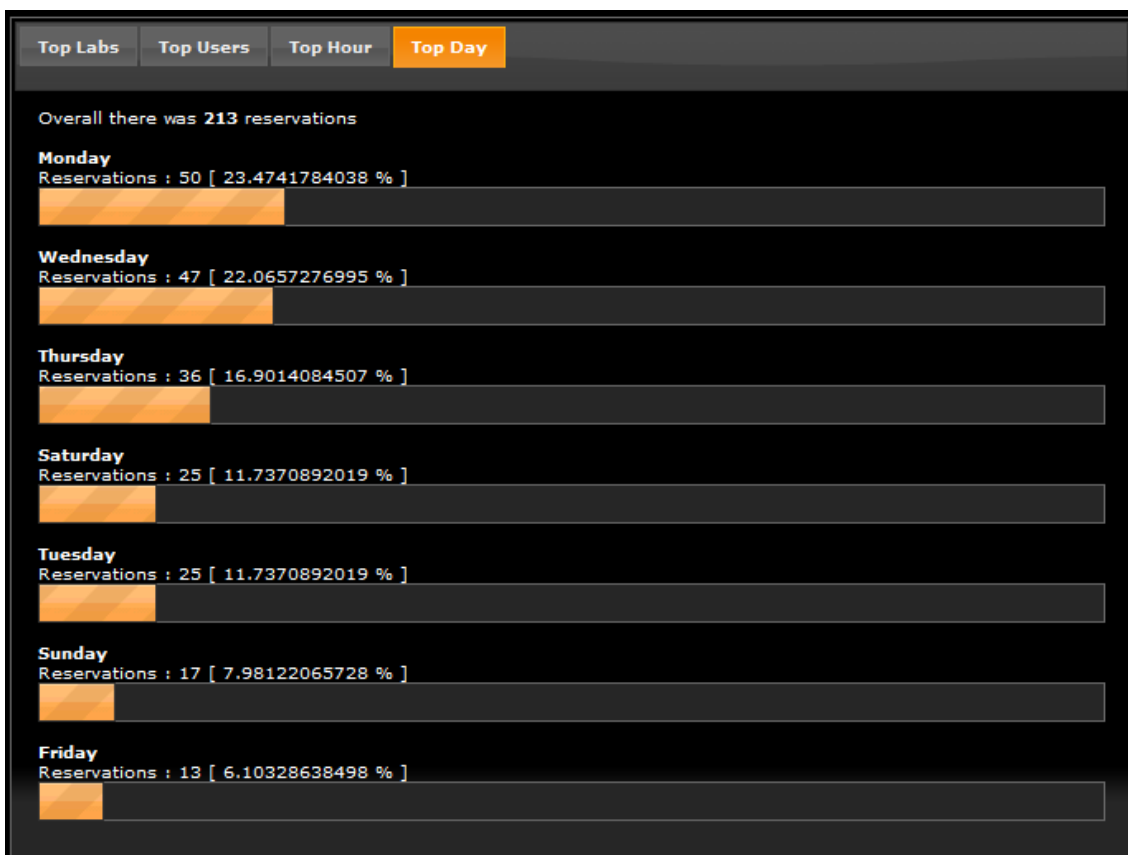
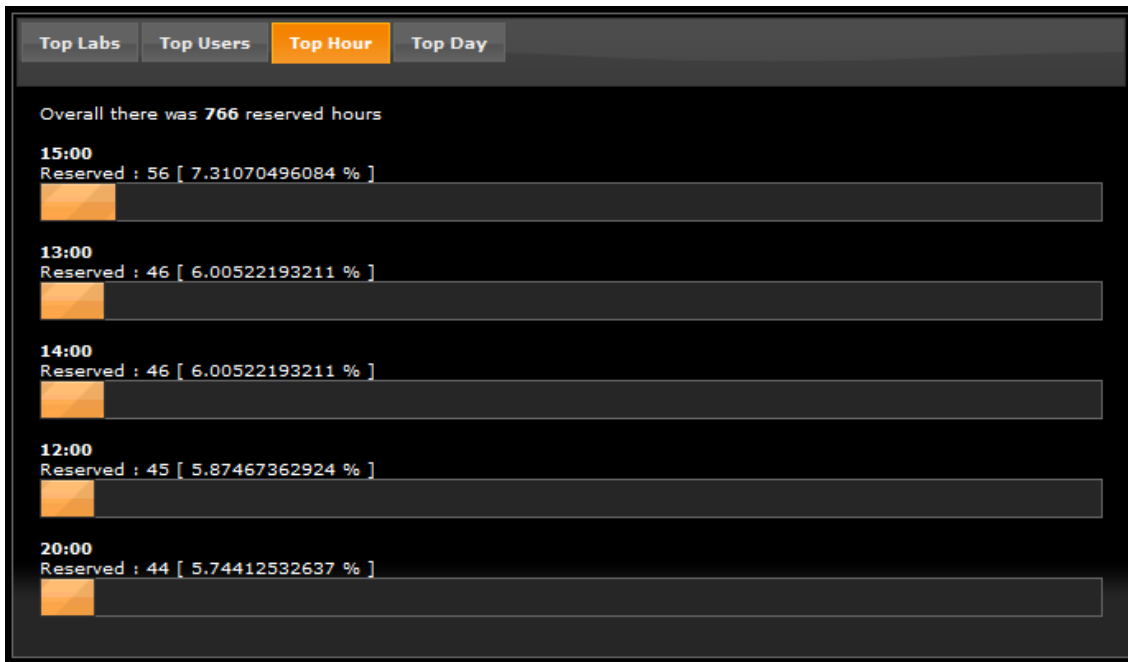
Ako vidno, tabuľka obsahuje 4 položky:

## Simulátor komunikácie v počítačovej sieti

- **TOP LABS** - zobrazuje prvých 5 najčastejšie simulovaných laboratórnych cvičení spolu s počtom spustení a základnými informáciami o cvičení.
- **TOP USERS** - zobrazuje prvých 10 používateľov, ktorý najčastejšie využívajú VLAB pre simuláciu laboratórnych cvičení. Pri každom z používateľov sa zobrazí počet simulácií a taktiež percentuálnu časť z celkového počtu všetkých simulácií.
- **TOP HOUR** - zobrazuje tie hodiny počas dňa, v ktorých si používatelia najčastejšie simulujú laboratórne cvičenia.
- **TOP DAY** - zobrazuje počet rezervácií v jednotlivých dňoch v týždni.



## Simulátor komunikácie v počítačovej sieti





## D. Inštalačná príručka

V tejto príručke je uvedený stručný postup, ako nainštalovať systém na nový počítač. VLAB beží na operačnom systéme Linux. Celý vývoj prebiehal na distribúcii Gentoo, ktorú odporúčame aj pre ďalších záujemcov o využívanie VLABu.

Minimálne hardvérové požiadavky závisia od množstva simulácií, ktoré by chcel používateľ spúšťať naraz. Odporúčané požiadavky (ktoré postačujú pre fungovanie systému aspoň v rozsahu, v akom bol otestovaný) sú nasledovné:

- Dvojjadrový procesor, frekvencia aspoň 2 GHz
- 2 GB pamäte RAM
- 10 GB voľného miesta na pevnom disku
- Operačný systém Linux, distribúcia Gentoo

Aby bol systém nainštalovaný správne je treba vykonať nasledujúce kroky:

- Stiahnuť z repozitára (alebo z webovskej stránky [http://sourceforge.net/project/showfiles.php?group\\_id=160317&package\\_id=230509&release\\_id=538693](http://sourceforge.net/project/showfiles.php?group_id=160317&package_id=230509&release_id=538693)) program Dynamips vo verzii 0.2.8-RC2 a nainštalovať ho. V Gentoo sa to celé vykoná príkazom „`emerge dynamips`“
- Nainštalovať VDE switch vo verzii 2.1.6, na ktorej bol náš projekt vyvíjaný.
- Na inštalačnom médiu sa nachádza adresár *opt*. V ňom sú umiestnené:
  - spúšťacie skripty jadra
  - program Dynagen s upravenými zdrojovými kódmi pre správne spúšťanie programu Pemu (Dynagen je vo verzii 0.11.0)
  - IOSy pre Cisco smerovače a firewally (*c3640-ik9s-mz.124-16.bin* pre smerovač Cisco 3640 a *pix803.bin* pre firewall Cisco PIX 525)
  - image so špeciálne upravenou distribúciou Linuxu pre UML host uzly v simuláciáchTento celý adresár je treba nakopírovať aj s jeho podpriechkami do koreňového adresára, takže na disku sa bude nachádzať adresár */opt*.
- V adresári so skriptami */opt/scripts* (dostaneme sa doň príkazom „`cd /opt/scripts`“) spustiť príkaz *make*. Ten skompiluje zdrojové súbory niektorých súčastí skriptov (tých, ktoré sú napísané v jazyku C a vyžadujú kompiláciu) a vytvorí tak binárne spustiteľné súbory
- Všetkým súborom s príponou *.sh* v adresári */opt/scripts* je treba nastaviť práva na spustenie. Vykoná sa to príkazom „`chmod 775 nazovsboru.sh`“. Tieto súbory predstavujú skripty v jazyku Bash.
- Pri simuláciách sa využíva hodnota *idlepc*, ktorá zabezpečí, aby bežiaci smerovač v idle stave nezaberali 100% výkonu procesora. Táto hodnota sa však mení v závislosti od konkrétneho hardvéru, na ktorom systém beží. Na zistenie jej hodnoty je treba spustiť skript „`/opt/scripts/idlepctest.sh`“. Tento skript spustí Dynamips a Dynagen, v ňom lab s jedným smerovačom bez *idlepc* hodnoty. Po nabehnutí Dynagen konzoly je treba počkať cca minútu na spustenie smerovača.

Potom sa spustením príkazu „idlepc get R1“ v tejto konzole vypočítajú idlepc hodnoty pre smerovač. Z nich je treba vybrať najlepšiu, zvyčajne najlepšie hodnoty označí Dynagen hviezdíčkou. Ak sa tak však nestane, alebo je takto označených viac čísel, najčastejšie bývajú najlepšie hodnoty okolo 50 a 60, najlepšie medzi. Ak takéto číslo nebude zobrazené, treba príkaz zopakovať. Po vybratí správnej hodnoty je vhodné skontrolovať, či naozaj klesla záťaž procesora procesom Dynamips (trebárs v programe Top). Ak áno, hodnotu idlepc je treba zapísať, pretože bude nutné ju vkladať do každej topológie pridávanej do databázy (a takisto už existujúce topológie upraviť, aby obsahovali novú idlepc hodnotu).

Po vykonaní týchto krokov by mala byť systémová časť projektu VLAB funkčná a pripravená na prepojenie s databázou a používateľskou časťou programu.

## **E. Článok do odborného časopisu**

## **F. Obsah priloženého CD**