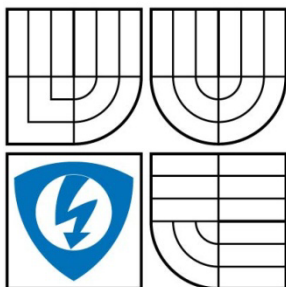


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

# AUTOMATICKÁ KONFIGURACE SÍŤOVÝCH PRVKŮ CISCO AKADEMIE

AUTOMATIC CONFIGURATION OF NETWORKING DEVICES IN CISCO ACADEMY  
LABORATORY

DIPLOMOVÁ PRÁCE  
DIPLOMA THESIS

AUTOR PRÁCE  
AUTHOR

BC. LUKÁŠ VERNER

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. DAN KOMOSNÝ, PH.D.

BRNO 2011



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Diplomová práce

magisterský navazující studijní obor  
Telekomunikační a informační technika

**Student:** Bc. Lukáš Verner

**ID:** 70014

**Ročník:** 2

**Akademický rok:** 2010/2011

## NÁZEV TÉMATU:

**Automatická konfigurace síťových prvků Cisco akademie**

## POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s dokumenty popisující činnost a zapojení laboratoře Cisco akademie FEKT, VUT v Brně. Vytvořte aplikaci pro automatickou konfiguraci síťových prvků Cisco s operačním systémem IOS (Internetwork Operating System). Aplikaci sestavte tak, aby zjednodušila přípravu laboratorních cvičení na přepínačích a směrovačích v laboratoři PA-128. Ovládání aplikace provedte pomocí webových stránek na výukovém serveru, který je umístěn v laboratoři. Vytvořený software přehledně zdokumentujte.

## DOPORUČENÁ LITERATURA:

- [1] GRAZIANI, R., JOHNSON, A. Routing Protocols and Concepts, CCNA Exploration Companion Guide. Cisco Press, USA, 2007. 606 s. ISBN 978-1-58713-206-3.
- [2] NEMETH, E., SNYDER, G., HEIN T. Linux - Kompletní příručka administrátora. Computer Press, 2004. 880 s. ISBN: 80-722-6919-4.
- [3] GILMORE, J. Velká kniha PHP5 & MySQL: Kompendium znalostí pro začátečníky i profesionály. [překlad Miroslav Kučera; RNDr. Jan Pokorný]. 1. vyd. Brno: Zoner Press, 2005. 711 s. ISBN 80-86815-20-X.

**Termín zadání:** 7.2.2011

**Termín odevzdání:** 26.5.2011

**Vedoucí práce:** doc. Ing. Dan Komosný, Ph.D.

**prof. Ing. Kamil Vrba, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Anotace**

Tato diplomová práce se zabývá návrhem a implementací systému pro automatickou konfiguraci síťových zařízení Cisco. Navržený systém se jmenuje *AutoConf* a je určen pro práci s konfiguračními soubory, určenými pro přepínače a směrovače, uloženými na konfiguračním serveru Eagle v učebně Cisco akademie. Ovládán je pomocí webových stránek umístěných na serveru Eagle.

Hlavní úlohou tohoto systému je umožnit uživatelům vytvářet nové laboratorní cvičení s konfiguračními soubory pro síťová zařízení, a pak je používat k automatické konfiguraci síťových zařízení. Seznamy laboratorních cvičení, jejich konfiguračních souborů a výsledky konfigurací jsou umístěny v relační databázi MySQL a je možné sledovat je na webových stránkách systému v přehledných tabulkách.

Výsledkem této diplomové práce by měl být systém pro automatickou konfiguraci síťových zařízení Cisco, usnadňující učitelům přípravu učebny na vypracování laboratorních úloh na přepínačích a směrovačích.

## **Klíčová slova**

AutoConf, síťové zařízení, směrovač, přepínač, startovací konfigurace, běžící konfigurace, webová stránka, webová aplikace, relační databáze, HTML, PHP, MySQL, SSH, TELNET, DHCP, EXPECT, EEM, TCL, BASH.

## **Abstract**

This diploma's thesis deals with proposal and system implementation for automatic configuration of Cisco network devices. This system is called *AutoConf* and it is intended for working with configuration files used for switches and routers. System *AutoConf* and its configuration files are stored on configuration server Eagle in Cisco academy laboratory. It is handled by web pages placed on server Eagle.

The main task of this system is to enable users to create new labs with configuration files for network devices, and then use the labs to automatically configure network devices. Lists of laboratory exercises, their configuration files and the results of configurations are stored in the MySQL relational database and can watch them on the website of the charts.

The result of this work should be a system for automatic configuration of Cisco networking devices, which helps teachers to facilitate the preparation of laboratory tasks on switches and routers.

## **Key words**

AutoConf, network device, router, switch, starting configuration, running configuration, web site, web application, relation database, HTML, PHP, MySQL, SSH, TELNET, DHCP, EXPECT, EEM, TCL, BASH.

VERNER, L. *Automatická konfigurace síťových prvků Cisco akademie*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 60 s. Vedoucí diplomové práce doc. Ing. Dan Komosný, Ph.D..

## Prohlášení

Prohlašuji, že svou diplomovou práci na téma AUTOMATICKÁ KONFIGURACE SÍŤOVÝCH PRVKŮ CISCO AKADEMIE jsem vypracoval samostatně pod vedením vedoucího diplomové práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne .....

.....

podpis autora

## **Poděkování**

Děkuji vedoucímu diplomové práce Ing. Danu Komosnému, Ph.D za velmi užitečnou metodickou pomoc a cenné rady při zpracování práce.

V Brně dne .....

.....

podpis autora

# Obsah

Úvod.....	10
1 Teoretický úvod.....	12
1.1 Popis učebny Cisco akademie .....	12
1.1.1 Směrovače .....	13
1.1.2 Přepínače .....	15
1.2 Konfigurace síťových zařízení s operačním systémem IOS.....	16
1.2.1 Operační systém Cisco IOS.....	17
1.2.2 Skriptovací jazyk TCL v operačním systému IOS.....	18
1.2.3 Detekce událostí v operačním systému IOS s využitím nástroje EEM.....	18
1.3 Operační systém Ubuntu .....	18
1.4 Vykonavatel Linuxových textových příkazů BASH .....	19
1.5 Nástroj pro automatizaci interaktivních aplikací EXPECT.....	19
1.6 Vzdálená komunikace mezi klientem a síťovým zařízením.....	20
1.6.1 Komunikační protokol TELNET .....	20
1.6.2 Bezpečná konfigurace síťových zařízení pomocí SSH .....	21
1.7 Sada programů platformy LAMP .....	21
1.7.1 Webový server Apache .....	21
1.7.2 Relační databáze MySQL.....	22
1.7.3 Programovací jazyk PHP .....	26
2 Automatická konfigurace síťových prvků Cisco Akademie .....	27
2.1 Základní funkce systému pro automatickou konfiguraci.....	27
2.2 Inicializace síťového zařízení systému pro automatickou konfiguraci.....	29
2.2.1 Průběh inicializace síťového zařízení.....	30
2.2.2 Instalace nového síťového zařízení .....	32
2.3 MySQL databáze systému pro automatickou konfiguraci .....	33
2.4 Webové moduly systému pro automatickou konfiguraci.....	37
2.4.1 Funkce pro připojení PHP do databáze MySQL.....	38
2.4.2 Funkce pro SSH spojení vykonavatele PHP příkazů s interpretem BASH.....	38
2.4.3 Autentizace uživatelů .....	39
2.4.4 Seznam zařízení pro automatickou konfiguraci .....	40
2.4.5 Seznam laboratorních úloh a konfiguračních souborů.....	41

2.4.6	Nahrávání konfiguračních souborů ze síťového zařízení na server Eagle .....	43
2.4.7	Vytváření a vykonávání konfigurací .....	45
2.4.8	Seznam konfigurací uložených v databázi systému .....	47
2.4.9	Modul pro rychlou konfiguraci síťového zařízení .....	47
2.5	Skripty pro terminál serveru Eagle .....	48
2.5.1	Komunikace serveru Eagle se síťovým zařízením pomocí EXPECT skriptu.....	48
2.5.2	Postup konfigurace síťového zařízení pomocí EXPECT skriptu .....	48
2.5.3	Skript pro změnu IP adresy a výchozí brány serveru Eagle.....	51
2.6	Instalace systému pro automatickou konfiguraci .....	53
2.6.1	Potřebné softwarové nástroje .....	53
2.6.2	Nastavení systému.....	53
1.1.1	Softwarové nástroje a verze použité při testování.....	54
3	Závěr.....	55
	Použitá literatura .....	56
	Seznam zkratk .....	58
	Seznam příloh.....	59

## Seznam obrázků

Obr. 1: Nákres učebny PA-128 – převzato z [14] .....	12
Obr. 2: Rozmístění přepínačů a směrovačů v učebně PA-128.....	13
Obr. 3: Schéma zapojení zařízení pro automatickou konfiguraci .....	28
Obr. 4: Průběh inicializace síťového zařízení .....	30
Obr. 5: Diagram databáze systému pro automatickou konfiguraci .....	33
Obr. 6: Přihlašovací okno systému pro automatickou konfiguraci .....	40
Obr. 7: Ukázka seznamu zařízení určených pro automatickou konfiguraci .....	41
Obr. 8: Ukázka tabulky úloh určených pro automatickou konfiguraci .....	42
Obr. 9: Ukázka zapojení síťových zařízení pro kopírování .....	43
Obr. 10: Formulář pro nahrávání konfiguračních souborů přímo ze síťového zařízení.....	44
Obr. 11: Vývojový diagram nahrávání konfiguračního souboru ze síťového zařízení na server Eagle.....	45
Obr. 12: Postup vytváření konfigurace v systému pro automatickou konfiguraci.....	46
Obr. 13: Ukázka konfigurace vytvořené v systému pro automatickou konfiguraci.....	47
Obr. 14: Ukázka seznamu existujících konfigurací .....	47
Obr. 15: Rychlá konfigurace síťových zařízení .....	48
Obr. 16: Vývojový diagram EXPECT skriptu pro konfiguraci síťového zařízení.....	49
Obr. 17: Změna IP adresy serveru Eagle.....	52

# Úvod

V laboratoři PA-128 Fakulty elektrotechniky a komunikačních technologií v budově na Purkyňově ulici se vyučuje předmět Cisco akademie. Nacházejí se zde různá síťová zařízení, která se studenti učí ovládat. Problémem však je, že každé z těchto zařízení je potřeba před cvičením nakonfigurovat tak, aby se studenti nemuseli zabývat základní konfigurací, ale mohli přímo řešit předmět jejich laboratorní úlohy. Připravit každé zařízení zvlášť pro danou úlohu není složité, ale časově náročné. Mnohem jednodušší by bylo nakonfigurovat všechna síťová zařízení z jednoho počítače najednou.

V této diplomové práci se zabývám vytvářením komplexního systému pro automatickou konfiguraci síťových zařízení. Automatická konfigurace bude ovládána pomocí sady skriptů umístěných na specifickém konfiguračním počítači v učebně, s názvem server Eagle. Činnost skriptu je ovládána pomocí webové aplikace rovněž umístěné na serveru Eagle. Proces automatické konfigurace síťového zařízení lze rozdělit do tří částí.

První část řeší problém připojení síťového zařízení k systému pro automatickou konfiguraci. Síťová zařízení jako přepínače a směrovače, mají po svém zapnutí standardně nastavený základní konfigurační soubor. To znamená, že nejsou schopny komunikovat s žádným jiným síťovým zařízením. Existuje však možnost nakonfigurovat je tak, aby se po zapnutí připojily do sítě se serverem Eagle. V případě, že bychom je nechtěli vzdáleně nakonfigurovat, by to bylo zbytečné. Za tímto účelem jsem se pokusil vytvořit inicializační mechanismus, založený na spojení se serverem Eagle. Pokud chce uživatel zařízení nakonfigurovat ze serveru Eagle stačí, když se dané síťové zařízení v základním nastavení připojí k přepínači, na kterém je připojen server Eagle. Následně dojde k automatické inicializaci síťového zařízení, při které síťové zařízení automaticky nastaví vše potřebné tak, aby mohlo komunikovat se serverem Eagle.

Druhou část tvoří webové rozhraní, které uživateli umožní, pomocí sady jednoduchých formulářů, vybrat si konfigurační soubory a přiřadit je síťovým zařízením. Následně bude uživatel moci provést konfiguraci vybraných síťových zařízení a její úspěšnost bude možné sledovat přímo na webových stránkách systému pro automatickou konfiguraci. Webové rozhraní umožňuje také spravovat seznamy síťových zařízení pro automatickou konfiguraci, vytvářet laboratorní úlohy a vkládat do nich konfigurační soubory pro síťová zařízení. Datovou strukturu webových stránek zde tvoří databáze umístěná na serveru Eagle. Obsahuje informace o všech síťových zařízeních, které se budou při automatické konfiguraci používat,

konfigurační soubory řazené do větších celků podle laboratorní úlohy, do které patří, záznamy o vykonaných konfiguracích a jejich úspěšnosti a také seznamy uživatelů a jejich práv v systému pro automatickou konfiguraci.

Třetí část se zabývá samotným procesem konfigurace síťových zařízení. Ten je vykonáván pomocí skriptů umístěných na serveru Eagle. Tyto skripty jsou spouštěny prostřednictvím webové aplikace, přičemž se jim předávají pouze parametry s informacemi o tom, jakému síťovému zařízení se má nahrát jaký konfigurační soubor. Skripty pak navážou spojení s daným síťovým zařízením a přikážou mu, aby si samo stáhlo vybraný konfigurační soubor ze serveru Eagle a použilo jej pro svou konfiguraci. Výsledky pokusů o konfiguraci a případné chyby vzniklé při konfiguraci, jsou pak skripty vráceny zpět do webové aplikace. Pokud dojde k úspěšné konfiguraci síťového zařízení, můžeme toto zařízení odpojit od serveru Eagle a zapojit jej do sítě podle schématu zapojení dané laboratorní úlohy.

Výsledkem této diplomové práce by měl být systém pro automatickou konfiguraci síťových zařízení Cisco, usnadňující učitelů přípravu učebny na vypracování laboratorních úloh na prepínačích a směrovačích. Měl by být ovládán pomocí webového rozhraní umístěného na serveru Eagle.

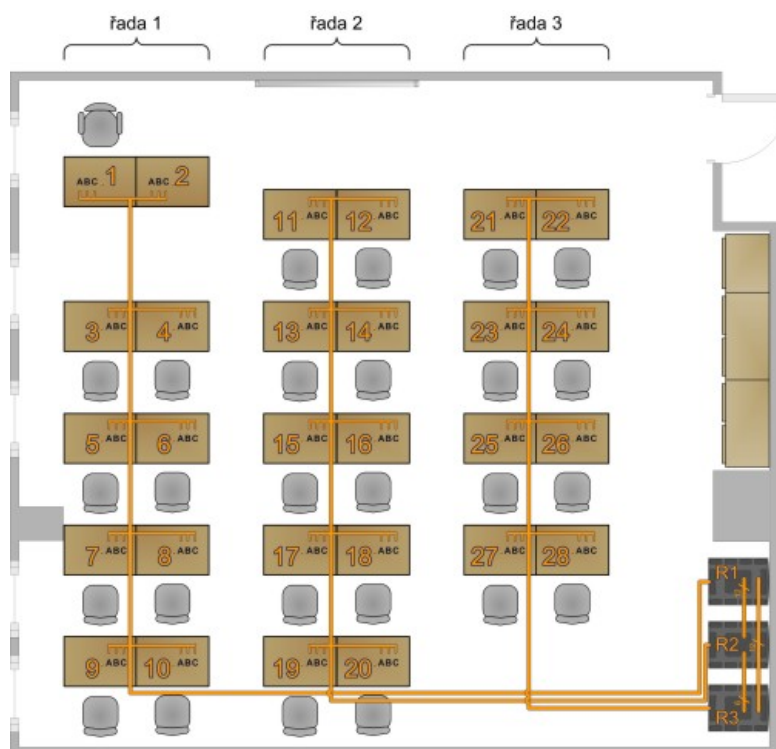
# 1 Teoretický úvod

## 1.1 Popis učebny Cisco akademie

Učebna PA-128 je laboratoř v budově fakulty elektrotechniky a komunikačních technologií Vysokého učení technického (dále jen VUT) na Purkyňově 118 v Brně. Je určena pro výuku kurzů CISCO NETWORKING ACADEMY. Jedná se o kurzy zaměřené na výuku síťových technologií, ovládání síťových zařízení, jakými jsou přepínače a směrovače.

V učebně se nachází 28 pracovních stanic s počítačem, přičemž dvě z nich jsou určeny pro učitele. Jedna z nich je i konfigurační server Eagle. Jsou na něm umístěny materiály pro výuku. V budoucnu na něm bude pracovat i systém pro automatickou konfiguraci zařízení Cisco.

Každý počítač v učebně je opatřen třemi rozhraními, dvěma síťovými kartami a jedním sériovým rozhraním. Pomocí UTP (Unshielded Twisted Pair) kabelů jsou tato rozhraní napojena k rozvaděčům v rohu místnosti (viz Obr. 1).

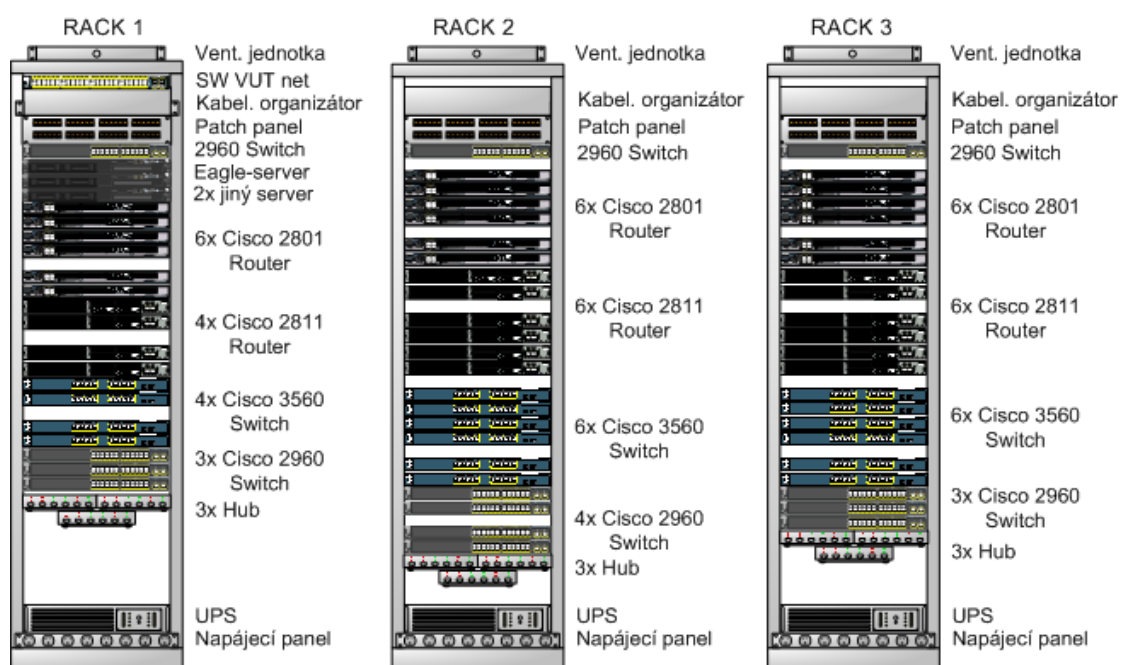


Obr. 1: Nákres učebny PA-128 – převzato z [14]

Na obrázku je vidět, že každá řada pracovních stanic má přidělen jeden rozvaděč. Podle čísla řady jsou rozvaděče označeny R1, R2 a R3. Každý rozvaděč obsahuje několik aktivních síťových prvků (přepínačů a směrovačů), které jsou připojeny na záložní zdroj

napětí pro reálné nasazení při výpadku napětí. Síťová rozhraní počítačových stanic jsou v rozvaděčích připojena k *patch panelům* a označena podle čísla stolu počítačové stanice a typu síťového rozhraní (viz Obr. 2). Síťová rozhraní počítačových stanic jsou v patch panelech značeny A, B, C a D. Jako A je označena síťová karta, která umožňuje přístup do sítě VUT. Rozhraním se značkou B a D se míní sériové rozhraní, jehož se využívá pro konfiguraci síťových zařízení pomocí konzolového připojení s využitím kabelu RJ45-DB9. Zbývající síťové rozhraní, tj. druhá síťová karta, je z vnější sítě zcela nedostupné, používá se při výuce, a má značení C.

Učebna PA-128 v současné době nabízí 18 směrovačů série 2801, 16 směrovačů série 2811, 13 přepínačů série 2960-24TT a 16 přepínačů série 3560v2-24PS-E. Tato aktivní síťová zařízení jsou podrobněji popsána níže [14].



Obr. 2: Rozmístění přepínačů a směrovačů v učebně PA-128

### 1.1.1 Směrovače

Směrovače patří mezi základní síťové prostředky v dnešních internetových sítích. Jejich základní úlohou je směrování a doručování *paketů*<sup>1</sup> uvnitř složité sítě. V záhlaví každého paketu zkoumají IP (Internet Protocol) adresy na *síťové vrstvě*<sup>2</sup> referenčního modelu OSI (Open System Interconnection) patřícího do standardů mezinárodní organizace ISO

<sup>1</sup> Sled informací, na úrovni síťové vrstvy, který je přenášen určitou sítí.

<sup>2</sup> Třetí vrstva referenčního modelu OSI. Protokoly této vrstvy obsluhují vyhledávání cesty a přenos dat v síti.

(International Standardization Organization). Na jejich základě se pak rozhodují, kam mají jednotlivé pakety poslat.

Mají však mnoho dalších důležitých funkcí, bez kterých by síť nemohla vůbec fungovat. Například:

- Směrovače mohou podporovat současné propojení více různých komunikačních protokolů jako Ethernet, ISDN (Integrated Services Digital Network), nebo Token Ring.
- Dokážou filtrovat nežádoucí síťový provoz, nebo jej rozdělit do izolovaných oblastí. Rozdělují síť do takzvaných *broadcastových domén*<sup>3</sup>.
- Můžou být také použity jako bezpečnostní bariéry, kontrolovat a řídit síťový provoz na základě přístupových oprávnění.
- Zajišťují spolehlivost a kvalitu síťových služeb, protože pro pakety znají od zdroje k cíli více cest a můžou z nich vybrat tu nejlepší.
- Automaticky zjišťují existenci, případně kolizi cest a přizpůsobují tomu mechanismy pro směrování paketů.

## Hardware směrovače

Hardware směrovačů by se dal rozdělit do tří základních skupin. Procesor směrovače je vykonavatel všech programů a funkcí potřebných pro chod směrovače. Další důležitou součástí směrovače jsou paměti směrovače, které se podle funkce, kterou plní, dělí na více druhů:

- RAM (Random-Access Memory) – dynamická operační paměť,
- NVRAM (Nonvolatile RAM) – paměť pro konfigurační soubor,
- ROM (Read-Only Memory) – obsahuje samozaváděcí program *bootstrap*<sup>4</sup>,
- Flash – paměť pro ukládání nových verzí operačního systému IOS (Internetwork Operating System).

Třetí skupinu hardwaru směrovačů tvoří porty. Jsou to síťová rozhraní pro propojení směrovače se sítí. Porty mohou být pevně spojené s hardwarem směrovače, nedají se ze směrovače odpojit, nebo mohou být tvořeny takzvanými zásuvnými moduly. Zásuvný modul je síťové rozhraní, které se zasouvá do skříně směrovače. Mezi nejčastější typy síťových

---

<sup>3</sup> *Broadcastová doména* je uzavřená síťová oblast v síti kde je možné všesměrové vysílání paketů.

<sup>4</sup> Pomocí programu *bootstrap* lze systém směrovače zavést do podoby minimálního konfiguračního souboru.

rozhraní, které se ve směrovačích používají, patří ethernetová rozhraní, sériová synchronní rozhraní, bezdrátová rozhraní nebo rozhraní pro připojení do sítě ISDN [1], [2], [13].

### **Směrovač série Cisco 2801 a 2811 umístěné v učebně PA-128**

Směrovače série Cisco 2801 a 2811 v učebně PA-128 se vyznačují vysokým množstvím poskytovaných síťových služeb a kvalitou bezpečnostního zabezpečení. Standardně obsahují dvě Fast-Ethernetová síťová rozhraní, rozhraní pro konzolové připojení a rozhraní AUX (Auxiliary port). Dále mají čtyři volné sloty pro zasunutí síťového rozhraní ve formě zásuvných modulů. Do jednoho z nich je vložen modul se dvěma sériovými synchronními rozhraními. Ty se často používají pro simulaci síťového WAN (Wide Area Network) spojení v laboratorních úlohách Cisco akademie. Ve směrovačích pracuje operační systém IOS, jež je vyladěn pro úkoly směrování.

#### **1.1.2 Přepínače**

Přepínače patří mezi moderní síťová zařízení pracující na úrovni *linkové vrstvy*<sup>5</sup> referenčního modelu OSI. Mají víc síťových rozhraní, portů, prostřednictvím kterých umožňují do sítě připojit víc síťových zařízení, například počítačů. Jejich inteligence spočívá v tom, že jsou schopny přepínat nebo filtrovat síťový provoz. Své rozhodnutí provádějí na základě *MAC*<sup>6</sup> (*Media Access Control*) adres připojených síťových zařízení. Přepínač pravidelně zkoumá, na jakém portu má připojené jaké síťové zařízení a zjištěné informace si ukládá do tabulky MAC adres. Následně je při přechodu *rámce*<sup>7</sup> přepínačem z jeho záhlaví přečtená MAC adresa cílového zařízení a porovnána s údaji v tabulce MAC adres. Po vyhodnocení je celý rámec odeslán pouze na port, ve kterém existuje síťové zařízení s určenou MAC adresou. Rámce se přitom nemění. Pokud má rámec neznámou nebo všesměrovou síťovou adresu, je zkopírován na všechny výstupní porty s výjimkou vstupního portu rámce.

---

<sup>5</sup> Druhá vrstva referenčního modelu OSI. Protokoly této vrstvy zajišťují komunikaci mezi sousedními uzly sítě.

<sup>6</sup> Hardwarová adresa síťové karty, případně síťového rozhraní.

<sup>7</sup> Sled informací, na úrovni linkové vrstvy, který je přenášen určitou sítí.

## Hlavní výhody použití přepínačů v sítích lze formulovat několika body:

- Větvení sítě na víc segmentů prostřednictvím připojení více síťových zařízení. Každý segment sítě je připojen na jiný port přepínače, můžeme mu definovat jinou šířku pásma.
- Vytváření menších kolizních domén – kolizní doména je definována množstvím síťových prvků v jednom segmentu sítě, vzájemně komunikujících na úrovni *fyzické vrstvy*<sup>8</sup> referenčního modelu OSI. Pokud začnou tyto prvky vysílat najednou, dojde ke kolizi a tím k nenávratnému poškození signálu.
- Možnost filtrace síťového provozu na základě MAC adresy a zabezpečení sítě proti připojení síťových zařízení s nepovolenou MAC adresou.
- Segmentace sítí do virtuálních sítí VLAN (Virtual Local Area Network) a tím snadnější správa a management uživatelských skupin sítě [2].

## Přepínače řady Catalyst 2960-24TT a Catalyst 3560v2-24PS-E

Tyto přepínače se v učeně PA-128 využívají pro výuku předmětů Cisco Akademie. Patří do řady síťových zařízení společnosti Cisco. Ve většině přepínačů společnosti Cisco pracuje operační systém IOS, vyladěný pro úkoly přepínání. Je tomu tak i v přepínači řady Catalyst 2960-24TT. Pro připojení do sítě obsahuje 24 Fast-Ethernetových rozhraní typu 100Base-TX a 2 rozhraní typu Gigabit-Ethernet 1000Base-T. Nevýhoda přepínače série 2960-24TT je, že nepodporuje konfiguraci pomocí EEM (Embedded Event Manager). To omezuje jeho funkci v systému automatické konfigurace, vytvořeném v této diplomové práci. Přepínač série 3560v2-24PS-E má novější operační systém IOS. Ten už konfiguraci pomocí EEM podporuje.

## 1.2 Konfigurace síťových zařízení s operačním systémem IOS

Konfigurace síťového zařízení je proces, při kterém uživatel komunikuje s operačním systémem daného síťového zařízení. Většina aktivních síťových zařízení společnosti Cisco se dá konfigurovat více způsoby. Například:

- pomocí konzolového připojení,
- využitím portu AUX pro vzdálenou konfiguraci síťového zařízení pomocí modemu,

---

<sup>8</sup> První vrstva referenčního modelu OSI. Zajišťuje vlastní přenos informací pomocí elektromagnetického signálu.

- pomocí protokolu TELNET (Telecommunication Network) nebo SSH (Secure Socket Shell) pro konfiguraci síťového zařízení přes terminál počítače,
- využitím konfigurace pomocí webového rozhraní.

Každé síťové zařízení má svůj vlastní konfigurační soubor. Ten je uložen v paměti NVRAM v podobě *startovací konfigurace* (startup-config). Do síťového zařízení se nahrává po jeho zapnutí. Konfigurace, pomocí které pak zapnuté síťové zařízení pracuje, se nazývá *běžící konfigurace* (running-config) a je uložena v dynamické paměti RAM.

Příkazy odeslané síťovému zařízení se vykonávají okamžitě, síťové zařízení není potřeba restartovat.

### 1.2.1 Operační systém Cisco IOS

V síťových zařízeních typu směrovač nebo přepínač se používá síťový operační systém IOS. Stejně jako ostatní operační systémy, IOS řídí hardwarové a softwarové zdroje daného zařízení, např. paměťovou alokaci, procesy, zabezpečení a systém souborů. Je to více-úlohový operační systém. Zahrnuje směrování, přepínání a další telekomunikační funkce [2].

Společnost Cisco nabízí spoustu typů operačního systému IOS. Tyto typy se odvíjejí hlavně od zařízení, ve kterém mají být použity, nebo od jejich hardwarových nároků, například zatížení procesoru a využití paměti.

Pro konfiguraci zařízení s operačním systémem IOS se nejčastěji používá příkazový řádek CLI (Command Line Interpreter), který poskytuje pevnou množinu víceslovných příkazů. Tato množina je určena módem, ve kterém se uživatel právě nachází. Ve směrovači rozlišujeme tyto základní módy:

- **Uživatelský mód** – povoluje uživateli sledovat stav zařízení, ale nedovoluje mu modifikovat konfiguraci.
- **Privilegovaný mód** – přístup do něj bývá chráněn heslem. Umožňuje měnit konfiguraci a sledovat aktuální konfiguraci v zařízení.
- **Globální konfigurační mód** – slouží přímo k zadávání příkazů pro základní konfiguraci.
- **Konfigurační mód síťového rozhraní** - poskytuje příkazy pro změnu konfigurace specifické pro dané rozhraní.
- **Konfigurační mód směrovacího protokolu** – slouží pro nastavení směrovacího protokolu a tím pádem i vytvoření směrovací tabulky [2], [4].

## 1.2.2 Skriptovací jazyk TCL v operačním systému IOS

IOS obsahuje TCL (Tool Command Language) shell, který je navrhnut k tomu, aby umožňoval spustit TCL příkazy přímo z příkazového řádku CLI. Pokud je TCLSH (Tool Command Language shell) povolen, můžeme zadávat příkazy TCL přímo v příkazovém řádku. Příkaz je po zadání odeslán na TCL interpret a pokud je rozpoznán, je okamžitě vykonán a výsledek je vyslán na výstup. TCL příkazy je možné zadávat předem do TCL skriptu, kterým je obyčejný textový dokument s koncovkou *tcl*. Skript je následně nahrán do paměti a spuštěn [6].

## 1.2.3 Detekce událostí v operačním systému IOS s využitím nástroje EEM

EEM je nástroj užívaný v operačních systémech IOS. Je určený k provádění akcí na základě monitorovaných událostí. Pokud monitorovaná událost nastane, EEM spustí EEM politiku, která je přiřazená k dané události. EEM politika může být definovaná pomocí EEM appletu nebo pomocí TCL skriptu.

EEM applety i TCL skripty slouží k definování akcí, které se mají provést, pokud je spuštěna EEM politika. Ovšem applety mají omezenější funkci než TCL skripty. Na rozdíl od EEM appletu mohou být v TCL skriptech definovány i další podmínky, cykly či větve, na základě kterých se může zařízení nakonfigurovat. TCL skript je ale nutno po vytvoření nahrát do zařízení jako samostatný soubor. EEM applet stačí nakonfigurovat v konfiguračním režimu appletu.

### Detektory událostí v EEM

Detektory událostí jsou mechanismy monitorující, zda událost nastala. Pokud monitorovaná událost nastane, spustí se EEM politika. Podle verze IOS se liší i použitelné detektory. Mezi nejpoužívanější patří například Application-Specific, CLI, Counter, Enhanced Object Tracking, Interface Counter, None, Resource, SYSLOG, System manager, Timer a Watchdog [1].

## 1.3 Operační systém Ubuntu

Ubuntu patří mezi jednu z mnoha distribucí operačního systému Linux. Tento operační systém je nainstalován na serveru Eagle v učebně Cisco akademie.

Linux je unixový operační systém. Skládá se z jádra operačního systému a systémových programů. Jádro systému řídí běh celého operačního systému. Umožňuje spouštění a běh aplikačních programů, přiděluje procesům hardwarové a jiné technické prostředky, například část paměti. Systémové programy využívají pro svou činnost služby volané jádrem operačního systému ve formě samostatných procesů.

Uživatel operačního systému Linux pracuje pouze se systémovými programy. Může je spouštět například pomocí zadávání textových příkazů do příkazového řádku systému.

## 1.4 Vykonavatel Linuxových textových příkazů BASH

BASH (Bourne Again Shell) je vykonavatel textových příkazů CLI. Tvoří rozhraní mezi operačním systémem a uživatelem. Je to druh shellu, který funguje na textové bázi. Mnohé své funkce převzal od předchozích shellů, například od Korn shellu a C shellu. Většinou je základním interpretem v mnoha linuxových distribucích. Oproti ostatním shellům má spoustu vylepšení. Například možnost edice příkazového řádku, historie příkazů s neomezenou délkou, řízení úloh, funkce a aliasy, indexovaná pole s neomezenou velikostí, a celočíselnou aritmetiku se základem 2 až 64 [3][5].

### Spouštění příkazů a skriptů v příkazovém interpretu BASH

Zadáme-li příkaz v příkazovém řádku, BASH nejprve zjistí, zda zadaný příkaz existuje, a následně existující příkaz vykoná. V proměnné *PATH* se nacházejí cesty do všech příkazových databází, které BASH má. Pokud chceme třeba spouštět náš skript z jakéhokoliv adresáře, stačí jen přidat cestu k němu do proměnné *PATH*.

Jestliže spouštíme skript, BASH nejprve vytvoří nový pod-proces. Ten běží jako sub-shell, čte řádky ze skriptu jeden po druhém, následně je interpretuje a vykoná. Po skončení skriptu skončí i sub-shell a systém se vrací do rodičovského shellu. BASH může obsahovat i takzvané vestavěné příkazy. Jejich použitím nedochází k vytvoření nového procesu.

## 1.5 Nástroj pro automatizaci interaktivních aplikací EXPECT

EXPECT je nástroj pro automatizaci interaktivních aplikací jakými jsou TELNET, FTP (File Transfer Protokol), atd. Pomocí něj je možné vyvinout automatizované zkušební skripty, zajišťující spolehlivost vykonávání operace. Nejčastěji se používá v příkazovém řádku CLI.

## Základní příkazy nástroje EXPECT:

Příkaz *spawn* se používá pro start procesu. Tímto příkazem se startuje jiný program. Prvním argumentem je jméno programu, který bude startovat. Následující argumenty jsou data umožňující start daného programu. Například pro start procesu FTP spojení se serverem *ftp.linux.net* lze použít příkaz:

```
spawn ftp ftp.linux.net
```

Příkaz *send* – pro posílání příkazů do procesu. Tento příkaz používá příkazy jiných programovacích jazyků a posílá je do procesu jako textové pole formátu *String*. Příkaz *send* posílá textový řetězec bez jakéhokoli formátování. Pro posunutí kurzoru na nový řádek je potřeba za textový řetězec vložit parametr „\n“. Následující příklad odešle do procesu slovo „Ahoj“ a posune kurzor na další řádek:

```
send "Ahoj\n"
```

Parametr „\r“ se používá pro odeslání textového řetězce ukončeného parametrem *ENTER*.

Příkaz *expect* čeká na odezvu z procesu. Nejčastěji *expect* čeká na speciální sled znaků, které souhlasí s parametrem příkazu. Pokud v procesu nastane chyba a příkaz odeslaný pomocí *send* nevrátí žádnou odpověď, příkaz *expect* pořád čeká na nějakou odpověď. Za tímto účelem má EXPECT definovanou speciální proměnnou *timeout*. Tato proměnná nastavuje čas, po který příkaz *expect* čeká na odpověď z procesu, a když ji nedostane, tak skončí. V následujícím příkladu příkaz *expect* čeká z procesu slovo „Ahoj“ a pokud jej dostane do 10 vteřin, vypíše na obrazovku: „Vítám Vás!“, jinak se vypíše: „Příště pozdravte!“

```
expect "Ahoj" {send "Vítám Vás!"}  
timeout {send "Příště pozdravte!"}
```

## 1.6 Vzdálená komunikace mezi klientem a síťovým zařízením

### 1.6.1 Komunikační protokol TELNET

Protokol TELNET představuje základní komunikační rozhraní pro vzdálenou konfiguraci síťového zařízení. Je definován normou RFC 854 a využívá se v síťové architektuře typu klient-server. Rozhraní typu příkazový řádek CLI, mu umožňuje přímé přihlášení do síťového zařízení a následně jeho konfiguraci nebo monitorování. Využití protokolu TELNET pro vzdálenou komunikaci se síťovým zařízením má dvě velké nevýhody způsobené tím, že komunikace mezi klientem a serverem není šifrována. Pokud útočník sleduje síťový provoz mezi klientem a serverem, může odposlechnout informace obsahující

uživatelské jméno a heslo využité při přihlašování protokolem TELNET. Dále může útočník přihlášenému klientovi ukrást spojení a odeslat na server libovolný příkaz. Anglicky se tento útok nazývá „*session hijacking*“[13].

## 1.6.2 Bezpečná konfigurace síťových zařízení pomocí SSH

SSH představuje spojovací protokol typu *klient-server* pro vzdálenou konfiguraci síťových zařízení. Využívá se pro šifrovaný přenos dat mezi klientem a síťovým zařízením po nedůvěryhodné síti.

Pro SSH spojení musí na serverové části běžet SSHD (Secure Shell Daemon) a na klientské části klient SSH. SSHD zajišťuje příjem požadavků odeslaných protokolem SSH. Dále testuje autentizaci z hlediska identity stroje i uživatele, a v případě platné autentizace vytváří šifrované spojení mezi klientem a serverem. Pro šifrování se používá dvojice šifrovacích klíčů RSA (Rivest Shamir Adelman), soukromý a veřejný klíč. Soukromý klíč je uložen na straně serveru a veřejný klíč na straně klienta. Klient odesílá data zašifrovaná pomocí veřejného klíče a server je proto dešifruje pomocí soukromého klíče. Tímto způsobem je zajištěno zabezpečení integrity dat odesílaných mezi klientem a serverem[13].

## 1.7 Sada programů platformy LAMP

LAMP (Linux Apache MySQL PHP server) je platforma, která představuje sadu instalačních programů. Tyto programy se používají pro implementaci dynamických webových stránek. Programy zahrnuté v platformě LAMP jsou volně šiřitelné.

LAMP zahrnuje technologie:

- Linux – operační systém,
- Apache – webový server,
- MySQL – relační databázový systém,
- PHP – skriptovací jazyk.

### 1.7.1 Webový server Apache

Apache je nástroj, který působí jako webový server. Jeho hlavní úlohou je zpracovávání požadavků, odesílaných webovými prohlížeči uživatelů, a zobrazení vyžádaných webových stránek pomocí kódu, umístěného ve vyžádaných souborech. Je to velmi výkonný nástroj obsahující mnoho funkcí.

Při tomto procesu uživatel zadá do webového prohlížeče adresu webové stránky, ta je pak odeslána přes internet na webový server. Webový server dotaz přijme, najde si informace na pevném disku a odešle odpověď ve formě zdrojového kódu a obrázku zpátky do prohlížeče uživatele. Prohlížeč pak zdrojový kód přeloží a informace zobrazí na obrazovce.

Zdrojový kód webových stránek přitom obsahuje mnoho značkovacích a skriptovacích jazyků. Například: HTML (HyperText Markup Language) pro tvorbu obsahu webových stránek, CSS (Cascading Style Sheets) pro stylování vzhledu webových stránek, JavaScript pro dynamizaci webových stránek na straně klienta, nebo PHP (PHP: Hypertext Preprocessor) pro dynamizaci webových stránek na straně serveru, atd [9].

### 1.7.2 Relační databáze MySQL

MySQL je relační databázový systém s architekturou klient/server. Systém sestává z databázového serveru MySQL a libovolného množství klientů. Klienti mohou být buď jednotliví uživatelé nebo aplikační programy, které komunikují se serverem. To znamená, že se dotazují na data, ukládají změny, nebo tvoří nové záznamy do tabulek databáze na serveru. Komunikovat s databází MySQL je možné přímo na serveru, ale také třeba pomocí internetu na dálku[8], [9], [10].

#### Návrh databáze

Prvním krokem tvorby databázových programů je návrh databáze. Ten pak bude mít vliv na funkčnost a efektivitu programu, náročnost programování, údržbu a flexibilitu. Návrh kvalitní databáze je složitý proces. Má mnoho pravidel a vyžaduje zkušenosti.

Při vytváření tabulek můžeme u každé tabulky specifikovat její typ. MySQL podporuje více typů tabulek, např. MyISAM, InnoDB, Heap. Pokud při vytváření tabulky neurčíme její typ, systém MySQL jej určí sám podle svého nastavení v konfiguračním souboru pod proměnnou *default-table-type*.

#### Použité datové typy MySQL

- **Integer:** 32 bitová celá čísla,
- **Varchar:** Textový řetězec proměnné délky, maximálně 65 536 znaků.
- **Blob:** Datový typ ukládající binární data o maximální délce  $2^{16} - 1$  bajtu.
- **Timestamp:** Datový typ pro ukládání data a časových údajů. Jeho sloupce jsou automaticky aktualizovány při každé změně záznamu v tabulce.

## Základní komponenty konceptuálního návrhu databáze

Konceptuální model se vytváří pomocí Entitně-relačního diagramu (ER). ER diagram je grafické znázornění všech entit, atributů, datových typů, relací primárních a cizích klíčů, které daná databáze obsahuje.

### Entity (Tabulky)

Entitou se rozumí tabulka, která obsahuje informace o jakékoliv události, věci, atd. Každá entita znázorněná v ER diagramu obsahuje řadu informací. To znamená, že pokud v naší databázi pro automatickou konfiguraci pojmenujeme entitu jménem *SEZNAM ZAŘÍZENÍ*, bude obsahovat seznam všech zařízení, spojených s naší databází.

### Atributy (Sloupce)

Atribut entity představuje jeden sloupec v dané entitě. Vyjadřuje vlastnosti každého záznamu v entitě. V každé entitě by se měl nacházet atribut, který by tvořil takzvaný *jedinečný identifikátor* entity. Jedinečný identifikátor má pro každou instanci v tabulce jedinečnou hodnotu, díky které lze každý záznam jednoznačně identifikovat. Jedinečný identifikátor může být složen i z více atributů. V entitě *SEZNAM ZAŘÍZENÍ* jím může být třeba název zařízení nebo identifikační číslo *ID*, pokud by mohla nastat situace, že by se jednotlivá zařízení jmenovala stejně.

### Relace

Relace popisují vzájemné vztahy entit dané tím, že data, která do databáze vkládáme, jsou navzájem nějakým způsobem vázána. V ER diagramu jsou relace znázorněny jako čáry propojující jednu, nebo více entit. Pro každou stranu relace se určuje *maximální kardinalita* vztahu entit, které daná relace propojuje. Je to nejvyšší počet záznamů jedné entity, které mohou být sdruženy s entitou na druhé straně relace. Maximální kardinalita může být rovna jedné (v ER diagramu je značena jednoduchou čarou), nebo více (v ER diagramu se značí vidličkou ve spoji s entitou). Dále se určuje *minimální kardinalita*, která určuje minimální počet záznamů dané entity, sdružených s entitou na opačné straně. Může být rovna nule (značí se prázdným kroužkem na konci čáry relace), jedné (značí se krátkou, kolmo vedenou čárkou na konci čáry relace). Kardinalita jedna a právě jedna se v ER diagramu značí pomocí dvou krátkých kolmých čárek na konci relace.

Kardinalita relace mezi entitami tvoří takzvanou *relační vazbu* mezi entitami. Podle typu relační vazby dělíme relace na:

- Relace typu jedna k jedné – je to vztah, kdy k záznamu jedné entity můžeme přiřadit maximálně jeden záznam entity druhé a naopak.
- Relace typu jedna k více – situace, kdy jeden záznam první entity může být přiřazen jednomu nebo více záznamům druhé entity, ale záznam druhé entity může být přiřazen maximálně jednomu záznamu první entity.
- Relace typu více k více – zvláštní situace, kdy záznamy obou entit můžou být přiřazeny žádnému, jednomu, nebo více záznamům na druhé straně relace. V praxi se vazba relace typu více k více nahrazuje vložení další nové entity doprostřed relace, čímž vznikají dvě relace s vazbou typu jedna k více.
- Rekurzivní relace – jsou relace, které vznikají mezi záznamy v jedné entitě. Mohou mít také vazbu jedna k jedné, jedna k více, nebo více k více [8], [9].

### **Primární a cizí klíče**

Primární a cizí klíče jsou speciální atributy používané v relačních databázích. Slouží pro snadnější vyhledávání záznamů v databázi a zachování integrity dat.

#### **Primární klíč**

Úkolem primárního klíče je co nejvíc urychlit vyhledávání záznamu v tabulce. Primární klíč musí být definován v každé entitě databáze. Může být složen z více atributů v dané entitě. Cizí klíč musí být unikátní a nesmí být prázdný. To znamená, že žádné dva záznamy v entitě nesmí mít stejný primární klíč. Pro rychlou práci databáze je důležitá kompaktnost primárního klíče. Díky tomu je snadnější zpráva záznamů v databázi protože primární klíče mohou být použity jako cizí klíče v jiných entitách databáze. Primární klíče jsou obvykle prezentovány číselným údajem.

#### **Cizí klíče**

Úkolem cizího klíče je odkazovat na záznamy v jiných entitách databáze. Při definici entit v databázi, nehrají cizí klíče velkou roli. Je to jen další sloupec v entitě. Využívají se ale při tvorbě inteligentních dotazů na výběr záznamů z databáze [8], [9], [10].

## Normalizační pravidla

Normalizační pravidla byla zavedena, aby zabránila situacím, které označujeme jako *Anomálie*:

- *Anomálie vkládání* – je to situace, kdy nelze do databáze vložit data kvůli umělé závislosti mezi sloupci v tabulce.
- *Anomálie odstraňování* – situace, kdy odstranění dat způsobí neúmyslnou ztrátu jiných dat.
- *Aktualizační anomálie* – situace, kdy změna jediného údaje v tabulce vyžaduje aktualizaci více řádků v tabulce.

### První normálová forma: Vyloučení opakovaných dat

Definice normálových forem není vždy pro normálního uživatele čitelná. Zde je stručné vysvětlení první normálové formy. Musíme eliminovat sloupce se stejným obsahem a vytvořit pro každý z nich další entitu. Každý takovýto záznam v nové entitě musí být jednoznačně identifikovaný primárním klíčem. Takto vzniklou novou entitu pak propojíme s původní entitou relací s vazbou jedna k více. Tím se primární klíč z nové entity stane cizím klíčem v entitě původní [11].

### Druhá normálová forma: Odstranění částečných závislostí

Hovoříme-li o databázi v druhé normálové formě, znamená to, že je v první normálové formě a všechny neklíčové atributy v každé entitě jsou funkčně závislé pouze na celém primárním klíči. To znamená, že pokud je atribut v entitě závislý jen na části primárního klíče v dané entitě, měla by se pro něj vytvořit nová entita s primárním klíčem, na kterém by byl atribut závislý jednoznačně. Druhá normálová forma se tedy týká pouze entit, které mají primární klíč složený s více atributů [11].

### Třetí normálová forma: Odstranění tranzitivních závislostí

Tranzitivní závislost atributů jedné entity je závislost atributu na jiném atributu ve stejné entitě, přičemž ani jeden z těchto atributů není primárním klíčem. Problém se řeší vytvořením nové entity, kam se přemístí funkčně závislé atributy a jeden z nich se stane primárním klíčem. Databáze splňuje třetí normálovou formu, pokud splňuje první a druhou normálovou formu a v žádné z jejich entit neexistuje tranzitivní závislost. Třetí normálová forma pokrývá 90 procent všech chyb, které jsme při návrhu databázového systému schopni

udělat. Proto pokud databáze v třetí normálové formě splňuje funkční hledisko systému, ve kterém bude pracovat, je její návrh považován za správný [11].

## Typy příkazů pro práci s daty v databázi

Typy příkazů pro práci s daty v databázi se dají rozdělit do několika skupin podle typu operací, které vykonávají:

- výběr záznamů z databáze (*SELECT*),
- vkládání záznamů do databáze (*INSERT*),
- změna záznamů v databázi (*UPDATE*),
- mazání záznamů z databáze (*DELETE*).

### 1.7.3 Programovací jazyk PHP

PHP patří mezi nepoužívanější skriptovací jazyky pro dynamizaci webových stránek. Pracuje na straně serveru. To znamená, že veškerá data jsou zpracovávána na serveru. Do uživatelského prohlížeče jsou posílány pouze výsledky. PHP patří mezi jazyky s otevřeným zdrojovým kódem, což je jeho největší výhodou.

Další předností PHP je, že jeho kód můžeme vkládat přímo do statických stránek HTML ve formě PHP skriptů. Aby takto vložený kód mohl pracovat, musí se stránka předat *engine PHP*<sup>9</sup>, ten je bude interpretovat. Kód PHP se od kódu odděluje pomocí určitých značek proto, aby mohl *parser*<sup>10</sup> okamžitě určit, které oblasti na stránce představují kód PHP. Prohlížeči webových stránek jsou po zpracování odesílány soubory, které neobsahují žádný zdrojový kód PHP [8].

---

<sup>9</sup> Engine PHP je vykonavatel příkazů skriptovacího jazyka PHP.

<sup>10</sup> Parser je mechanismus, který má za úkol rozeznat příkazy PHP ve zdrojovém kódu webových stránek.

## 2 Automatická konfigurace síťových prvků Cisco Akademie

### 2.1 Základní funkce systému pro automatickou konfiguraci

V této diplomové práci se zabývám vytvořením komplexního systému pro automatickou konfiguraci síťových prvků Cisco s operačním systémem IOS. Jeho hlavní úlohou bude usnadňování přípravy laboratorních cvičení na přepínačích a směrovačích v laboratoři PA-128 Fakulty elektrotechniky a komunikačních technologií Vysokého učení technického v Brně. Výhodou přitom bude snadné ovládání – prostřednictvím webových stránek uložených na konfiguračním serveru Eagle (dále jen Eagle), umístěném v laboratoři. Celý systém jsem pojmenoval *AutoConf*. Název *AutoConf* je zkratkou spojení *Automatic Configuration* (Automatická konfigurace).

Základní funkce systému *AutoConf* je vzdálená automatická konfigurace síťových zařízení. Tato funkce a všechny ostatní funkce, které systém *AutoConf* nabízí, se zajišťují pomocí webových stránek. Díky nim může uživatel sledovat stav spojení serveru Eagle s jednotlivými síťovými zařízeními. Dále může do systému vkládat nové laboratorní úlohy a konfigurační soubory pro daná síťová zařízení. Ty je pak možné přiřadit připojeným síťovým zařízením a následně je tímto způsobem konfigurovat.

Samotná konfigurace je realizována pomocí skriptů EXPECT. Pomocí nich se provádí telnetové připojení serveru Eagle k síťovým zařízením a následná konfigurace síťových zařízení. EXPECT skript odesílá příkazy síťovému zařízení a na základě jeho odpovědi pak reaguje odesláním dalších příkazů, nebo ukončením programu. Po úspěšném provedení konfigurace pomocí EXPECT skriptu je dané síťové zařízení nakonfigurováno pro měření laboratorní úlohy. Stačí jej už jen odpojit od sítě serveru Eagle a zapojit podle schématu zapojení dané laboratorní úlohy.

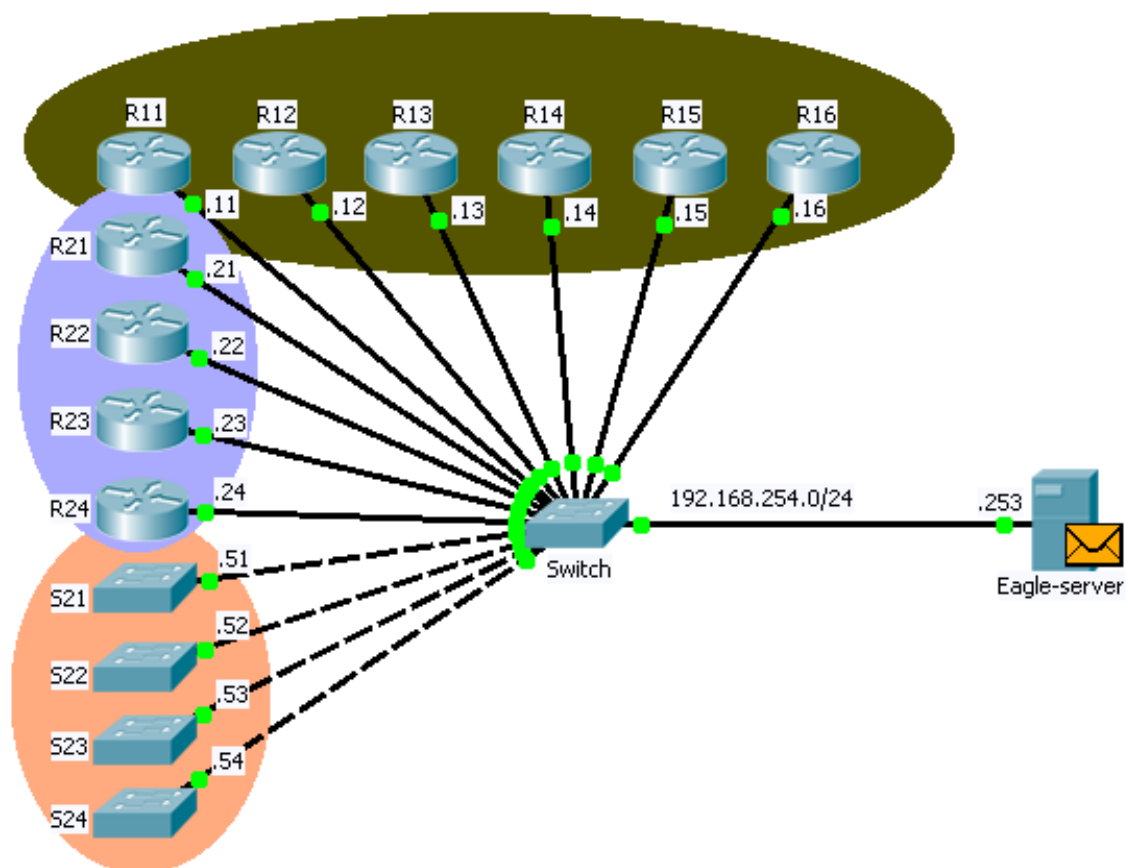
Všechny informace o připojených síťových zařízeních, laboratorních úlohách, případně vykonaných či nevykonaných konfiguracích, jsou přitom spravovány pomocí PHP funkcí a databáze MySQL. Konfigurační soubory jednotlivých laboratorních úloh se nacházejí přímo na serveru Eagle.

Aby bylo možné ovládat síťová zařízení pomocí systému *AutoConf*, musí být tato zařízení fyzicky připojená do sítě serveru Eagle. Obr. 3 ukazuje zjednodušené schéma zapojení sítě serveru Eagle využívané pro konfiguraci síťových zařízení systémem *AutoConf*.

Toto zapojení zobrazuje zapojení síťových zařízení pouze jednoho rozvaděče. Pro připojení síťových zařízení z dalších rozvaděčů je využita stejná *síťová topologie*<sup>11</sup> s tím rozdílem, že propojovací přepínač v každém dalším rozvaděči je propojen se serverem Eagle pomocí spojení s *hlavním přepínačem serveru Eagle*<sup>12</sup>.

V každém síťovém zařízení, použitém v systému *AutoConf*, nastává při jeho startu proces inicializace. Při tomto procesu se v síťovém zařízení spustí operační systém IOS a nastaví se běžící konfigurace. Systém *AutoConf* nabízí možnost automatického nastavení síťového zařízení pro komunikaci se systémem automatické konfigurace. Toto nastavení je možné aplikovat na připojená síťová zařízení pomocí akčních tlačítek ve webové aplikaci systému *AutoConf*.

Přístup na webové stránky systému *AutoConf* je zabezpečený pomocí definice uživatelů a uživatelských práv.



**Obr. 3: Schéma zapojení zařízení pro automatickou konfiguraci**

<sup>11</sup>Tvar fyzického propojení síťových zařízení.

<sup>12</sup>Přepínač, na který je přímo připojen server Eagle. Na Obr. 3 značen jako *Switch*.

## 2.2 Inicializace síťového zařízení systému pro automatickou konfiguraci

Inicializace síťového zařízení v systému *Autoconf* je proces při kterém dochází k automatické konfiguraci síťového zařízení tak, aby byla možná komunikace mezi serverem Eagle a tímto zařízením. To umožní konfigurovat síťové zařízení pomocí webových stránek systému *Autoconf*.

Pro konfiguraci síťového zařízení ze serveru Eagle, musí na něm být nastaveno alespoň jedno síťové rozhraní pro komunikaci se serverem Eagle. Mezi tímto rozhraním a serverem Eagle musí existovat fyzické spojení. Dále musí být povolena a nastavená možnost konfigurace síťového zařízení pomocí protokolu TELNET a také je třeba mít nastavené heslo pro přechod do privilegovaného režimu síťového zařízení. IP adresa rozhraní připojeného k serveru Eagle, heslo pro autorizaci spojení pomocí protokolu TELNET a heslo pro přechod do privilegovaného režimu jsou uloženy v databázi systému *AutoConf*.

Při startu síťového zařízení se standardně do jeho běžící konfigurace nahrává startovací konfigurace uložená v NVRAM. Pokud síťové zařízení nemá uloženou startovací konfiguraci, hledá ji v síti. Obvykle na TFTP serveru. Pokud žádný konfigurační soubor není nalezen, síťové zařízení se spustí s základní konfigurací a je nutné jej konfigurovat ručně pomocí konzolového nebo AUX spojení.

Pro inicializaci síťového zařízení v systému *AutoConf* je server Eagle využit jako takzvaný *Autoinstalační server*. Pokud tedy připojíme libovolný ethernetový port směrovače nebo přepínače do sítě, ve které se nachází server Eagle, dojde k automatické inicializaci síťového zařízení v systému *AutoConf*. Musí ale být splněny následující dvě podmínky:

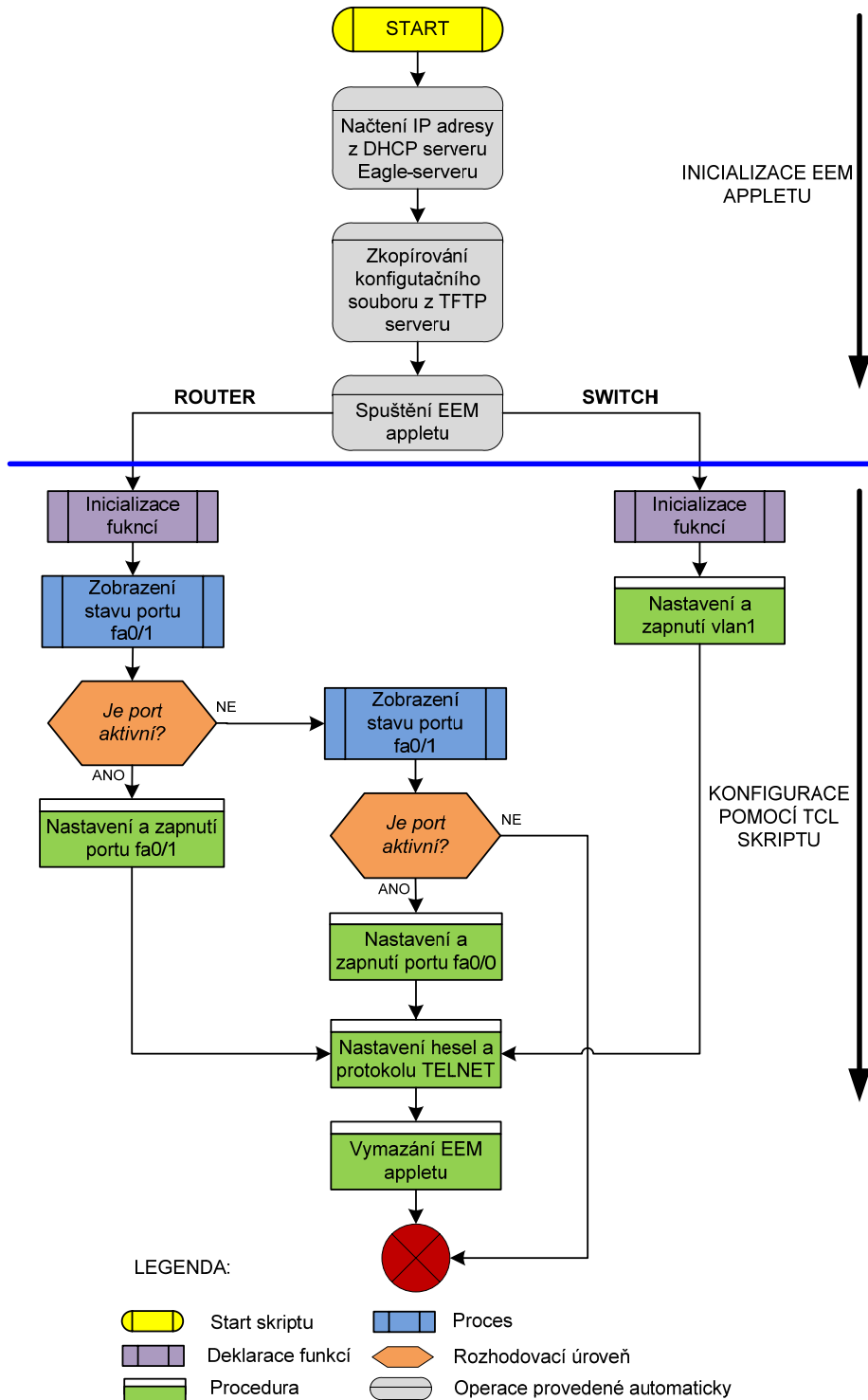
- Síťové zařízení nesmí obsahovat startovací konfiguraci a jeho běžící konfigurace nesmí být nijak změněna. Například pomocí konzolového nebo AUX spojení.
- Síťové zařízení musí mít nastavený standardní *konfigurační registr*<sup>13</sup> na hodnotu 0x2102 nebo 0xF. To mu umožní automatické vyhledávání konfiguračních souborů v NVRAM nebo v síti.
- V paměti FLASH síťového zařízení musí být předem uložený TCL skript *autoconf.tcl*.

---

<sup>13</sup> 16-bitové slovo paměti NVRAM, které určuje odkud se bude zavádět IOS, zda a odkud se bude načítat konfigurace, rychlost komunikace síťového zařízení s terminálem a další parametry.

## 2.2.1 Průběh inicializace síťového zařízení

Průběh inicializace síťového zařízení lze rozdělit do dvou fází. Inicializaci EEM appletu a konfiguraci síťového zařízení pomocí TCL skriptu *autoconf.tcl*. Grafické znázornění celkové inicializace je popsané vývojovým diagramem na Obr. 4.



Obr. 4: Průběh inicializace síťového zařízení

## Inicializace EEM appletu

Pokud síťové zařízení splňuje výše uvedené podmínky a připojíme ho do ethernetové sítě serveru Eagle, na jeho ethernetovém portu dojde k přidělení síťové IP adresy z DHCP (Dynamic Host Configuration Protocol) serveru nainstalovaného na serveru Eagle.

Dále se síťové zařízení automaticky pokouší stáhnout konfigurační soubor z TFTP serveru s IP adresou serveru Eagle. IP adresa, na kterou posílá požadavek je adresa získaná z DHCP serveru pomocí přednastaveného parametru *routers*<sup>14</sup>.

Poté, co dojde ke stažení konfiguračního souboru s názvem *network-confg*<sup>15</sup> ze serveru Eagle, do běžící konfigurace síťového zařízení se nastaví EEM applet *Autoconf*. Ten po deseti vteřinách spustí TCL skript *autoconf.tcl* uložený v paměti FLASH síťového zařízení. EEM applet *Autoconf* je zobrazen v následujícím zdrojovém kódu.

```
scripting tcl enddir flash:
# Specifikace cílové cesty externích kódovacích souborů TCL
# do paměti flash.
event manager applet Autoconf
# Vytvoření appletu s názvem Autoconf.
event timer watchdog time 10
# Nastavení detektoru spuštění appletu.
action 1.0 cli command "enable"
# Příkaz na cli: Přechod do privilegovaného režimu.
action 1.1 cli command "tclsh flash:autoconf.tcl"
# Spuštění TCL skriptu
action 2.0 cli command "end"
# Ukončení EEM appletu.
```

## Konfigurace síťového zařízení pomocí TCL skriptu *autoconf.tcl*

Hlavní úlohou TCL skriptu *autoconf.tcl* je konfigurace síťového zařízení tak, aby bylo možné jeho ovládání z webových stránek serveru Eagle. Každé síťové zařízení musím mít uložený svůj vlastní TCL skript protože obsahuje informace jako je IP adresa a hesla pro komunikaci s daným síťovým zařízením. Na základě IP adresy je později dané síťové zařízení identifikovatelné v systému *AutoConf*.

Prvním krokem konfigurace pomocí TCL skriptu *autoconf.tcl* je nastavení IP adresy. U přepínače stačí nastavit IP adresu pro *VLAN1*<sup>16</sup>, bylo možné síťové spojení se serverem Eagle. U směrovače je nutné IP adresu nastavit na port, který je připojen do sítě. Proto TCL skript u směrovačů testuje, který port je zapnutý a na ten pak nastaví požadovanou

---

<sup>14</sup> Parametr, který specifikuje seznam IP adres pro směrování dat v podsíti.

<sup>15</sup> Konfigurační soubor, který v síti standardně vyhledává síťové zařízení bez startovací konfigurace v NVRAM.

<sup>16</sup> VLAN, do které patří všechny porty přepínače v základním nastavení. Standardně se používá po management.

IP adresu. Potom se u obou druhů síťových zařízení natavuje protokol TELNET a heslo pro přechod do privilegovaného režimu. Nakonec se vymaže spouštěcí EEM applet, aby už nedocházel k jeho opětovnému spouštění.

### 2.2.2 Instalace nového síťového zařízení

Proces inicializace síťového zařízení v systému *Autoconf* je možný pouze u síťových zařízeních, na kterých operační systém IOS podporuje EEM applety a TCL skripty.

V současné době se v učebně PA-128 nachází následující druhy síťových zařízení:

- Směrovač *Cisco Router Catalyst 2801*: podporuje TCL a EEM.
- Směrovač *Cisco Router Catalyst 2811*: podporuje TCL a EEM.
- Přepínač *Cisco Switch Catalyst 3560v2-24PS-E*: podporuje TCL a EEM.
- Přepínač *Cisco Switch Catalyst 2960-24TT*: podporuje TCL, nepodporuje EEM.

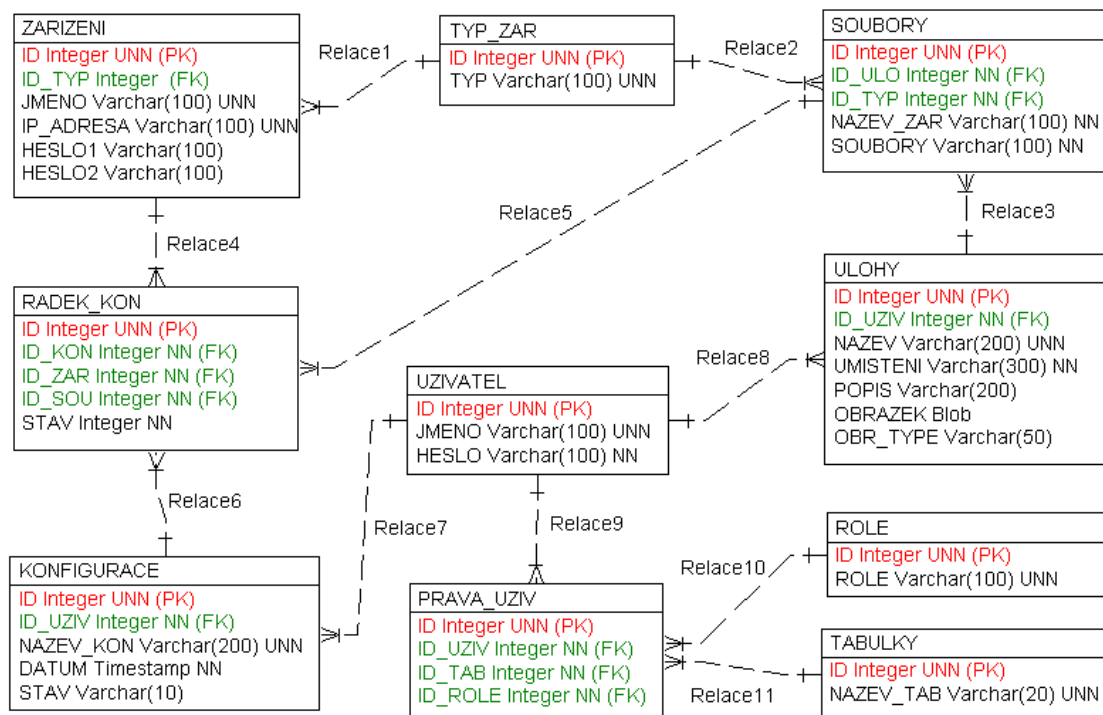
Protože přepínač *Cisco Switch Catalyst 2960-24TT* nepodporuje EEM applety, není možná jeho inicializace v systému *AutoConf*.

## 2.3 MySQL databáze systému pro automatickou konfiguraci

Systém *AutoConf* obsahuje relační databázi MySQL s názvem *Autoconf* pracující na serveru Eagle. Její hlavní úlohou je uchovávání záznamů potřebných pro fungování automatické konfigurace ze strany serveru Eagle. Obsahuje čtyři skupiny datových struktur, rozdělené do deseti entit:

- Záznamy o zařízeních, které se budou konfigurovat pomocí serveru Eagle.
- Seznam úloh a konfiguračních souborů, které je možné použít pro automatickou konfiguraci.
- Tabulky konfigurací. Tabulka konfigurací je struktura záznamů, kde jsou k vybraným zařízením ze seznamu zařízení přiřazeny konfigurační soubory vybraných úloh.
- Záznamy o uživateli a právech, které mají uživatelé v systému *Autoconf*.

Struktura databáze *Autoconf*, všech entit, relací, atributů, datový typů, primárních a cizích klíčů je zobrazena na Obr. 5. Primární klíče jsou značeny zkratkou *PK* (Primary key), cizí klíče se značí zkratkou *FK* (Foreign key), unikátní atributy zkratkou *U* (Unique) a atributy, které musí být vyplněny zkratkou *NN* (NOT NULL). Význam jednotlivých atributů ve všech entitách databáze *Autoconf* je popsán v tabulce (Tab. 2).



Obr. 5: Diagram databáze systému pro automatickou konfiguraci

Entita *ZARIZENI* představuje seznam všech směrovačů a prepínačů, určených pro automatickou konfiguraci. Dále obsahuje data potřebná pro vytvoření spojení s daným zařízením. Cizí klíč *ID\_TYP*, převzatý prostřednictvím *Relace:1*, odkazuje na typ zařízení určený atributem *TYP* v entitě *TYP\_ZAR*.

Entita *SOUBORY* obsahuje informace o konfiguračních souborech, určených pro automatickou konfiguraci. Každý soubor má přiřazen typ zařízení pomocí cizího klíče, atributu *ID\_TYP*.

Díky tomu, že entity *ZARIZENI* a *SOUBORY* mají přiřazený typ, můžeme pomocí *PHP* funkce zajistit, aby nebylo možné při konfiguraci přiřadit prepínači konfigurační soubor určený pro směrovač a opačně. Směrovači nejde přiřadit konfigurační soubor určený pro prepínač.

Entita *ULOHY* tvoří tabulku se záznamy o úlohách, které budou studenti měřit. Kromě názvu úlohy je zde možné zadat také krátkou poznámku o úloze a vložit obrázek se schematickým znázorněním úlohy. Primární klíč *ID* entity *ULOHY*, je prostřednictvím *Relace:3* přenášen do entity *SOUBORY*, kde definuje, do jaké úlohy je daný soubor přiřazen.

*Relace:3* také zajišťuje to, že pokud je libovolná úloha v entitě *ULOHY* vymazána nebo upravena, budou také záznamy v entitě *SOUBORY*, patřící k dané úloze, patřičně editovány, případně smazány.

Informace o jednotlivých konfiguracích jsou uloženy v entitách *KONFIGURACE* a *RADEK\_KON*. Každá konfigurace, zapsaná v databázi *Autoconf*, má svůj název, je definovaná určitým uživatelem, má informaci o tom, kdy byla naposled editována nebo vykonána, a obsahuje pole *STAV*, které indikuje informaci o tom, zda byla vykonána a na kolik procent byla úspěšná. Tato data jsou uložena v entitě *KONFIGURACE*. V entitě *RADEK\_KON* jsou uloženy pouze identifikátory zařízení spojené s identifikátory souborů, které budou v dané konfiguraci vykonávány.

Entita *UZIVATEL* spravuje informace o uživateli, jejich přihlašovací jména a hesla. V další entitě *TABULKY* je definován seznam tabulek, ve kterých se budou řešit uživatelská práva. Entita *ROLE* určuje seznam všech možných práv v systému. Entita *PRAVA\_UZIV* je konfiguračním souborem, prostřednictvím kterého se v *PHP* přiřazují funkce přihlášeným uživatelům mé webové aplikace pro automatickou konfiguraci zařízení *Cisco*. Jeden záznam v této entitě definuje, který uživatel (*ID\_UZIV*) má povolenou jakou roli (*ID\_ROLE*) v jaké tabulce (*ID\_TAB*).

Pro účely fungování systému *AutoConf* v učebně PA-128 jsem vytvořil dva uživatele s rozdílnými uživatelskými právy. Vyučující se budou do systému přihlašovat pomocí přihlašovacího jména *Učitel*, a studenti pomocí přihlašovacího jména *Student*.

V Tab. 1 jsou definována všechna práva uživatelům *Ucitel* a *Student*. Uživatel, který je přihlášen jako *Ucitel*, má v entitě *PRAVA\_UZIV* definována všechna možná uživatelská práva. Pracuje tedy jako administrátor systému a má přístup ke všemu, co systém *AutoConf* nabízí. Uživatel *Student* má uživatelská práva omezena pouze na editaci úloh a vytváření nových konfigurací. Nesmí editovat entitu se seznamem zařízení, může však nastavovat síťová zařízení. Smí vykonávat pouze konfigurace uložené v databázi.

**Tab. 1: Uživatelská práva uživatelů *Učitel* a *Student***

Role	Tabulka				Uživatel
	ZARIZENI	ULOHY	RADEK_KON	KONFIGURACE	
1. Mazat	Ano	Ano	Ano	–	ID: 1.
2. Přidat	Ano	Ano	–	–	
3. Upravit	Ano	Ano	Ano	–	Login: Ucitel
4. Nová konfigurace	–	–	Ano	Ano	
5. Vykonat	–	–	Ano	–	Heslo: Teacher
6. Prohlednout	Ano	Ano	Ano	Ano	
7. Rychlá konfigurace	–	–	–	Ano	
8. Nastav zařízení	Ano	–	–	–	
1. Mazat	Ne	Ano	Ne	–	ID: 2.
2. Přidat	Ne	Ano	–	–	
3. Upravit	Ne	Ano	Ne	–	Login: Student
4. Nová konfigurace	–	–	Ano	Ano	
5. Vykonat	–	–	Ano	–	Heslo: Student
6. Prohlednout	Ano	Ano	Ano	Ano	
7. Rychlá konfigurace	–	–	–	Ne	
8. Nastav zařízení	Ano	–	–	–	

**Tab. 2: Popis atributů v entitách databáze systému pro automatickou konfiguraci**

Entita	Atribut	Význam atributu
ZARIZENI	ID ID_TYP JMENO IP_ADRESA HESLO1 HESLO2	Primární klíč – identifikuje záznam v databázi. Cizí klíč – identifikátor typu zařízení. Jméno zařízení pro automatickou konfiguraci. IP adresa rozhraní, na které se bude server připojovat. Heslo pro připojení pomocí protokolu <i>TELNET</i> . Heslo pro přechod do privilegovaného režimu.
ULOHY	ID ID_UZIV NAZEV POPIS OBRAZEK OBR_TYPE UMISTENI	Primární klíč – identifikuje záznam v databázi. Cizí klíč – identifikuje uživatele, který úlohu vytvořil. Název úlohy. Popis úlohy, nebo poznámka k úloze. Bitová data obrázku v úloze. Typ obrázku, potřebné pro zobrazení obrázku. Umístění konfiguračního souboru na serveru Eagle.
SOUBORY	ID ID_ULO ID_TYP NAZEV_ZAR SOUBORY	Primární klíč – identifikuje záznam v databázi. Cizí klíč – definuje úlohu, do které soubor patří. Cizí klíč – identifikátor typu zařízení. Pojmenování zařízení v dané úloze. Název konfiguračního souboru.
TYP_ZAR	ID TYP	Primární klíč – identifikuje záznam v databázi. Definuje typ zařízení (směrovač, nebo přepínač).
RADEK_KON	ID ID_KON ID_ZAR ID_SOU STAV	Primární klíč – identifikuje záznam v databázi. Cizí klíč – definuje konfiguraci, které záznam patří. Cizí klíč – identifikátor zařízení z entity <i>ZARIZENI</i> . Cizí klíč – identifikátor souboru z entity <i>SOUBORY</i> . Stav konfigurace. (0 – nevykonána, 1 – vykonána).
KONFIGURACE	ID ID_UZIV NAZEV_KON DATUM  STAV	Primární klíč – identifikuje záznam v databázi. Cizí klíč – identifikuje uživatele, který úlohu vytvořil. Pojmenování konfigurace. Datum definuje čas vytvoření, nebo konfigurace dané konfigurace. Určuje stav konfigurace ( <i>Nová</i> – zatím nevykonána, <i>Změněná</i> – upravená, <i>Číslo v procentech</i> – procentuální úspěšnost konfigurace).
UZIVATEL	ID JMENO HESLO	Primární klíč – identifikuje záznam v databázi. Přihlašovací jméno uživatele. Přihlašovací heslo uživatele.
TABULKY	ID NAZEV_TAB	Primární klíč – identifikuje záznam v databázi. Jméno tabulky.
ROLE	ID ROLE	Primární klíč – identifikuje záznam v databázi. Definuje roli. Například: <i>mazání, úpravy, konfigurace</i> .
PRAVA_UZIV	ID ID_UZIV ID_TAB ID_ROLE	Primární klíč – identifikuje záznam v databázi. Cizí klíč – identifikátor uživatele. Cizí klíč – definuje tabulku, ve které je daná role možná. Cizí klíč – definuje roli, jakou má uživatel.

## 2.4 Webové moduly systému pro automatickou konfiguraci

Webové stránky systému *AutoConf* jsou složeny z PHP skriptů. Můžeme je rozdělit do několika skupin podle funkcí, které plní:

- PHP skripty pro připojení a komunikaci s databází MySQL: *db\_spojeni.php* – připojení k databázi MySQL, *db\_funkce.php* – skript obsahující základní funkce s dotazy typu *SELECT* pro načtení dat z databáze MySQL.
- PHP skripty pro komunikaci s interpretem BASH: *ssh\_spojeni.php* – připojení k interpretu BASH serveru Eagle pomocí SSH, *bash.php* – spouští skript *tester.exp* pro test spojení a kontrolu nastavení mezi serverem Eagle a síťovým zařízením, *inicializace.php* a *bashrychle.php* – spouštějí skript *configurator.exp* pro nahrání konfiguračních souborů do síťového zařízení, *flash.php* – spouští skript *flash.exp* pro nahrání TCL skriptu do paměti FLASH síťového zařízení a *device.php* – spouští skripty *ipchange.exp* a *getconfig.exp* pro nahrávání konfiguračních souborů ze síťových zařízení na server Eagle.
- PHP skripty pro kontrolu autorizace uživatele v systému AutoConf: *autorizace.php* – skript pro přihlášení uživatele do systému, *kontrola.php* – skript s funkcí pro kontrolu uživatelských přístupových práv do jednotlivých modulů.
- Skripty pro zajištění vzhledu webových stránek: *styly.css* – stylový soubor, *hlavicka.php* – vytváření hlavičky, *nabidka\_vlevo.php* – postranní nabídky stránek.
- Skripty společných PHP funkcí: *php\_funkce.php* – obsahuje funkce, které jsou zapotřebí ve více než jednom PHP skriptu, například funkce pro vytváření postranní nabídky akčních tlačítek.
- Webové moduly, které mají úlohu zprostředkovat vizuální obsah stránek: *zarizeni.php*, *ulohy.php*, *konfigurace.php*, *seznam\_kon.ph*. Obsahují funkce pro zobrazení vizuálního obsahu webových stránek systému *AutoConf*. Jsou zabezpečeny skriptem pro kontrolu autorizace uživatele *kontrola.php* a funkcí *php\_pristup()* pro kontrolu přístupu daného uživatele.
- Javascriptové moduly s funkcemi vykonávanými prohlížečem webových stránek na straně klienta: *form.js* – Obsahuje funkce pro kontrolu formulářů na straně klienta. *Prototype.js* – Knihovna, která umožňuje použití funkcí technologie AJAX (Asynchronous JavaScript and XML). Pomocí těchto funkcí lze měnit obsah webových stránek bez nutnosti jejich opětovného načtení. Knihovna je převzatá z [15].

## 2.4.1 Funkce pro připojení PHP do databáze MySQL

Pro připojení PHP do databáze jsem vytvořil samostatný skript z názvem *db\_spojeni.php*. Zdrojový kód popsany v následujících řádcích představuje základní funkci pro připojení PHP skriptu k databázi MySQL. Při instalaci systému *AutoConf* je nutno v tomto souboru editovat proměnné se síťovou adresou serveru databáze a přihlašovacími údaji do databáze.

```
function pripojit_k_db() {
// Jméno uživatele pro připojení do databáze.
$mysqluzivatel = „jmeno_uzivatele“;
// Heslo pro připojení do databáze.
$mysqlheslo = „heslo“;
// IP adresa serveru MySQL.
$mysqlhost = „adresa_serveru“;
// Název databáze.
$mysqldb = „Autoconf“;
// Vytvoření nového konstruktora spojení s databází MySQL serveru.
$mysqli = @new mysqli($mysqlhost, $mysqluzivatel, $mysqlheslo,
$mysqldb);
// Podmínka navázání spojení.
if(mysqli_connect_errno()) {
    echo „<p>Nepodařilo se navázat připojení
    s databází MySQL</p>“;
    echo „</div></div></body></html>\n“;
    exit();
}
// Vrácení konstruktora spojení k MySQL.
return $mysqli;
}
```

Funkce *pripojit\_k\_db()* vrací konstruktor spojení PHP s databází *Autoconf* potřebný pro odesílání dotazů na databázi. Funkce *new mysqli* otevírá nové spojení se serverem MySQL. Funkce *mysqli\_connect\_errno()* vrací kód chyby posledního pokusu a spojení s databází.

## 2.4.2 Funkce pro SSH spojení vykonavatele PHP příkazů s interpretem BASH

Pro spuštění BASH a EXPECT skriptů využívá PHP interpreta BASH, ke kterému se připojuje pomocí SSH. Za účelem vytvoření SSH spojení mezi vykonavatelem PHP příkazů a terminálem serveru Eagle byla vytvořena funkce *ssh2\_pripoj()*, která vrací konstruktor spojení PHP s SSH klientem serveru. Zdrojový kód na následující straně popisuje její průběh.

Pokus o vytvoření SSH spojení pomocí funkce *ssh2\_connect* se může opakovat dvakrát po sobě. Toto řešení jsem zvolil kvůli stabilitě systému *AutoConf*. Pro zobrazení jedné

webové stránky je funkce `ssh2_pripoj()` často volána více než 10 krát. To způsobuje občasné nezdařilé vykonání funkce `ssh2_connect`, a proto je nutné ji opakovat.

Funkce `ssh2_auth_password()` odesílá do interpretu přihlašovací jméno a heslo potřebné pro autorizaci ssh spojení. Její vstupní parametry `login` a `heslo` je nutné při instalaci systému *AutoConf* nastavit na uživatele, který má administrátorská práva spojená s adresářovou strukturou systému *AutoConf*. Tím se zajistí to, aby bylo možné spustit BASH a EXPECT skripty.

```
function ssh2_pripoj() {
// Cyklus umožňující dva pokusy o navázání SSH spojení.
for($pocitadlo=0; $pocitadlo<"2"; $pocitadlo++){
// Kontrola existence spojení PHP s SSH.
if(!($connection = @ssh2_connect("localhost", 22)) and
$pocitadlo == "1"){
// Výpis a ukončení funkce pokud se nepodaří navázat "
// spojení ani na druhý pokus o SSH spojení.
echo "Chyba spojení SSH! /n";
exit();
} else $pocitadlo++;
}
// Kontrola přihlášení uživatele.
// Zde je potřebné vyplnit správné přihlašovací jméno a heslo.
if(!@ssh2_auth_password($connection, "login", "heslo")){
// Výpis a ukončení pokud se autorizace nepodaří.
echo "Chyba autentifikace SSH! \n";
exit();
}
// Vrácení konstruktoru spojení k SSH.
return $connection;
}
```

### 2.4.3 Autentizace uživatelů

Autentizace uživatelů je v systému *AutoConf* zavedena nejen kvůli zabezpečení systému proti neoprávněnému přístupu a editaci dat, ale také pro definici uživatelských práv uživatelů systému.

#### Přihlášení uživatele

Skript *prihlasit.php* je vstupním přihlašovacím formulářem pro přístup uživatele do systému *AutoConf* (viz Obr. 6). Údaje odeslané tímto formulářem jsou vyhodnoceny skriptem *autorizace.php*. Skript *autorizace.php* porovná přihlašovací jméno a heslo s údaji uloženými v databázi v entitě *UZIVATEL*. Pokud jméno a heslo souhlasí, skript uloží do proměnné `$_SESSION` informace o přihlášeném uživateli a přesměruje jej do modulu *zarizeni.php*.

Login:	Ucitel
Heslo:	●●●●●●●●
<input type="button" value="Přihlásit"/>	

**Obr. 6: Přihlašovací okno systému pro automatickou konfiguraci**

Při neplatném přihlášení, odhlášení uživatele, nebo vypršení doby nečinnosti uživatele v systému, je uživatel automaticky přesměrován do modulu *prihlasit.php*.

### **Kontrola autorizace uživatele ve webových modulech systému**

Skript *kontrola.php* zabezpečuje kontrolu uživatelského přístupu do webových modulů volaných uživatelem. Obsahuje funkci *php\_pristup(\$tabulka, \$role)* se dvěma vstupními parametry. Na základě obsahu parametrů *\$tabulka*, *\$role* a identifikátoru přihlášeného uživatele funkce detekuje, zda má uživatel přístup do daného PHP skriptu.

Kontrola autorizace se vykonává před zpřístupněním dat z každého PHP skriptu systému *AutoConf*. Ověřuje se, zda má přihlášený uživatel přístupová práva na prohlížení daného modulu. Pokud se uživatel pokusí vstoupit do modulu, na který nemá přístupová práva, je automaticky odhlášen a přesměrován do modulu *prihlasit.php*.

#### **2.4.4 Seznam zařízení pro automatickou konfiguraci**

Seznam přepínačů a směrovačů připojených k systému *AutoConf* je uložen v databázové entitě *ZARIZENI*. Pro editaci seznamu konfigurovatelných zařízení jsem vytvořil modul *zarizeni.php*.

Základní úlohou skriptu *zarizeni.php* je zobrazení informací o síťových zařízeních určených pro automatickou konfiguraci (viz Obr. 7). Sloupec *Stav* je dynamicky měnící se pole, obsahující informace o aktuálním stavu spojení s daným síťovým zařízením. Pro testování stavu síťového zařízení se ve skriptu *zarizeni.php* volá EXPECT skript *tester.php* (viz strana 48). Ten se pokouší připojit na dané síťové zařízení pomocí protokolu TELNET a otestovat na něm platnost všech hesel.

Tab. 3 popisuje seznam všech možných stavů zařízení definovaných v systému *AutoConf* nejen v modulu *zarizeni.php*, ale také v modulech, kde probíhá konfigurace síťových zařízení pomocí systému *AutoConf*.

Ve sloupci *Možnosti* jsou akční tlačítka pro práci s daným zařízením. Pokud je uživatel přihlášen jako administrátor, tedy *Učitel*, může zařízení smazat, případně editovat jeho nastavení. Uživatel *Student* má uživatelská práva pouze pro tlačítko *TCL*. Akční tlačítko *TCL*

## SEZNAM ZAŘÍZENÍ

Typ	Název	IP adresa	Telnet heslo	# heslo	Stav	Možnosti
Router	Rack1-R11	192.168.254.11	cisco	cisco	<i>Připojeno!</i>	Upravit Smazat ? TCL
Router	Rack1-R12	192.168.254.12	cisco	cisco	<i>Připojeno!</i>	Upravit Smazat ? TCL
Router	Rack1-R13	192.168.254.13	cisco	cisco	<i>Připojeno!</i>	Upravit Smazat ? TCL
Router	Rack1-R14	192.168.254.14	cisco	cisco	<i>Testuji...</i>	Upravit Smazat ? TCL
Router	Rack1-R15	192.168.254.15	cisco	cisco	<i>Odpojeno!</i>	Upravit Smazat ? TCL
Switch	Rack1-S21	192.168.254.51	cisco	cisco	<i>Připojeno!</i>	Upravit Smazat ? TCL

**Obr. 7: Ukázka seznamu zařízení určených pro automatickou konfiguraci**

kopíruje TCL skript *autoconf.tcl* do paměti FLASH na síťová zařízení. Funkce skriptu *autoconf.tcl* byla popsána v kapitole 2.2. Pod odkazem v otazníku se skrývá návod jak registrovat síťové zařízení v systému *AutoConf*.

**Tab. 3: Seznam všech možných stavů síťového zařízení v systému pro automatickou konfiguraci**

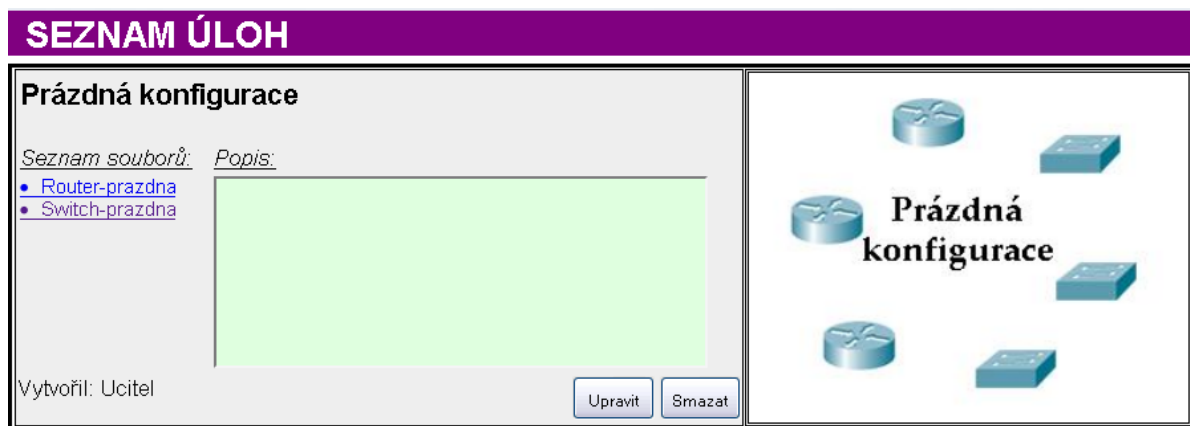
Výstup	Vysvětlení významu výstupu skriptu <i>tester.exp</i>
<i>Odpojeno!</i>	Zařízení není připojeno nebo má jinou IP adresu.
<i>Nastavuji!</i>	Zařízení se momentálně inicializuje pro komunikaci.
<i>Chybné heslo TELNET!</i>	Přístup TELNET byl odmítnut kvůli špatnému heslu.
<i>Chybné heslo “#“ !</i>	Špatné heslo pro přístup do privilegovaného režimu.
<i>Připojeno!</i>	Zařízení je připojeno a připraveno ke konfiguraci.
<i>Hotovo!</i>	Úspěšné vykonání operace, například nahrání konfigurace.
<i>Chyba odesílání!</i>	Chyba při nahrávání konfiguračního souboru do síťového zařízení.
<i>Chyba konfigurace!</i>	Chyba při konfiguraci startovací konfigurace přepínače.

### 2.4.5 Seznam laboratorních úloh a konfiguračních souborů

Práci s úlohami a jejich konfiguračními soubory zajišťuje PHP modul *ulohy.php*. Generuje HTML kód se seznamem úloh, stará se o edici nových úloh, mazání a upravování celých úloh, nebo jen jejich konfiguračních souborů.

Obr. 8 ukazuje výstup funkce *tabulka\_ulo()*, volané skriptem *ulohy.php*. Pokud je uživatel přihlášen jako *Student*, má podle nastavení uživatelských práv (viz Tab. 1) možnost měnit záznamy v tabulce *SEZNAM ÚLOH*. V zájmu ochrany úloh vytvořených uživatelem

*Učitel*, má *Student* možnost upravovat nebo mazat úlohy vytvořené pouze uživatelem *Student*. Uživatel *Učitel* pracuje jako administrátor, a proto může měnit všechny záznamy v tabulce úloh.



Obr. 8: Ukázka tabulky úloh určených pro automatickou konfiguraci

### Postup při návrhu nové laboratorní úlohy

Před návrhem nové laboratorní úlohy musíme mít nejdříve na klientském počítači uloženy konfigurační soubory pro novou úlohu. Poté se přihlásíme do webových stránek systému *AutoConf* a v modulu *ulohy.php* klikneme na akční tlačítko *Přidat další*. Zobrazí se nám formulář pro editaci laboratorních úloh.

Edice laboratorní úlohy má dvě části. V první části se editují záznamy v entitě *ULOHY* databáze MySQL. Povinný parametr *Název úlohy* a nepovinné parametry *Popis* a *Schéma zapojení úlohy*. Ve druhé části se do systému vkládají konfigurační soubory pro síťová zařízení definovaná v dané úloze. Uživatel má možnost vkládat konfigurační soubory buď přímo ze svého počítače, nebo kopírováním ze síťového zařízení (viz kapitola 2.4.6). Do databáze se ukládá pouze název konfiguračního souboru. Samotné konfigurační soubory jsou ukládány na server Eagle do adresáře *SOUBORY/Název laboratorní úlohy*.

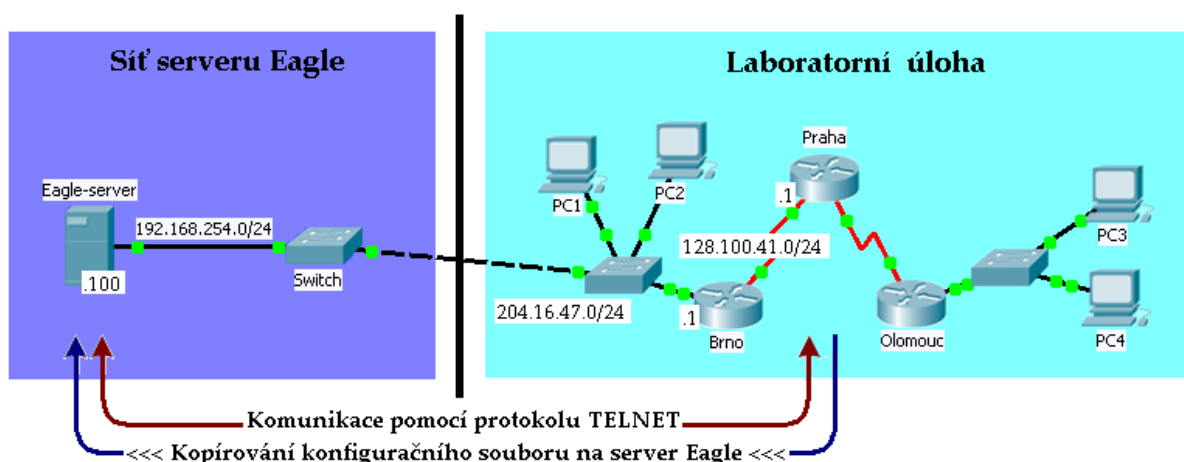
### Mazání laboratorních úloh a konfiguračních souborů

Při mazání laboratorních úloh a konfiguračních souborů v modulu *ulohy.php* se mažou i záznamy z entity *RADEK\_KON*, které obsahují identifikátor vymazaného konfiguračního souboru z laboratorní úlohy. To může způsobit existenci prázdných konfigurací v entitě *KONFIGURACE*. Prázdná konfigurace definovaná v entitě *KONFIGURACE* neobsahuje žádné záznamy v entitě *RADEK\_KON*.

Proto je při každém mazání konfiguračního souboru, případně laboratorní úlohy, volána funkce *kontrola\_konfiguraci()* pro kontrolu prázdných konfigurací v entitě *KONFIGURACE*. Pomocí ní jsou tyto konfigurace mazány.

#### 2.4.6 Nahrávání konfiguračních souborů ze síťového zařízení na server Eagle

Pro snadnější vysvětlení principu nahrávání konfiguračního souboru ze síťového zařízení na server Eagle uvedeme příklad kopírování konfiguračního souboru ze směrovače *Praha* ze síťového zapojení laboratorní úlohy na Obr. 9.



Obr. 9: Ukázka zapojení síťových zařízení pro kopírování

Pro jeho úspěšné provedení musí být splněny následující kroky:

- **Vytvoření TCP/IP spojení mezi síťovým zařízením a serverem Eagle**

TCP/IP spojení zajistíme například tak, že pomocí ethernetového kabelu propojíme přepínač sítě serveru Eagle s přepínačem laboratorní úlohy. Síť *128.100.41.0/24* musí ale být dostupná ze sítě *204.16.47.0/24*.

- **Možnost zabezpečeného připojení na síťové zařízení pomocí protokolu TELNET**  
Na směrovači musí být povoleno připojení pomocí protokolu TELNET s použitím hesla.
- **Správné vyplnění všech parametrů formuláře pro nahrávání konfiguračních souborů přímo ze síťového zařízení (viz Obr. 10).**

**Nový konfigurační soubor ze síťového zařízení**

Typ zařízení:

Název zařízení:  Počet pokusů:

**Nastavení Eagle-serveru jako Host v LAN síťového zařízení:**

IP adresa:  Maska:  Brána:

**Parametry síťového zařízení:**

IP adresa:

Heslo pro Telnet spojení:  Odpojeno!

Heslo pro přechod do # režimu:

**Obr. 10: Formulář pro nahrávání konfiguračních souborů přímo ze síťového zařízení**

Během procesu kopírování dochází na krátký čas ke změně IP adresy ethernetového rozhraní serveru Eagle. Tato se mění proto, aby bylo možné vytvořit TCP/IP spojení mezi serverem Eagle a síťovým zařízením. V prvním kroku uživatel propojil síť serveru Eagle se sítí laboratorní úlohy. Proto musí následně být do formuláře zadána vhodná IP adresa a maska, která bude přidělena serveru Eagle. Musí být taková, aby patřila do připojené sítě a nekolidovala s IP adresami ostatních zařízení v dané síti.

Výchozí brána je IP adresa rozhraní směrovače, na kterou se budou odesílat data směřující mimo síť, ve které je připojen server Eagle. Je to nepovinný parametr, ale je vhodné, aby byl zadáván, pokud nekopírujeme konfigurační soubor z přímo připojeného síťového zařízení.

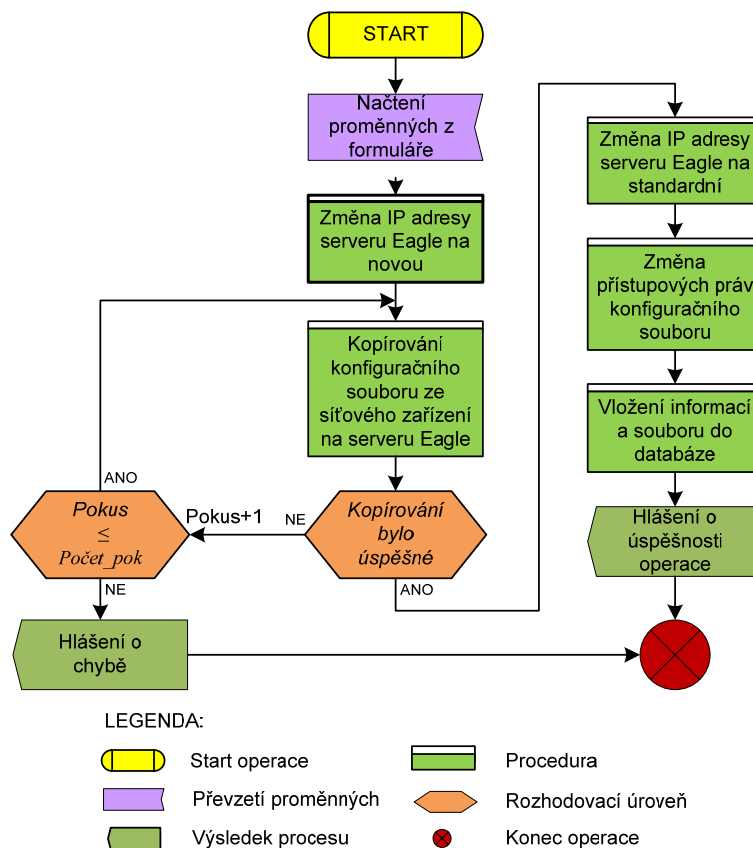
Dále je nutné zadat správné parametry síťového zařízení, ze kterého se bude konfigurační soubor kopírovat. Jsou jimi IP adresa dostupného síťového rozhraní, heslo pro přihlášení pomocí protokolu TELNET a heslo pro přechod do privilegovaného režimu.

Někdy, obzvláště při kopírování souborů ze vzdálenějších síťových zařízení, se proces kopírování neprovede hned na první pokus. Tento jev je způsobeno delší dobou aktualizace *ARP(Address Resolution Protocol) tabulek*<sup>17</sup> a konvergence celé sítě. Tehdy je proto vhodné zvýšit počet pokusů kopírování. Tím se sice prodlouží doba probíhání operace, ale zvýší se pravděpodobnost úspěšného kopírování.

<sup>17</sup> Tabulka obsahující záznamy MAC adres odpovídajících IP adresám síťových zařízení.

Obr. 11 je vývojovým diagramem průběhu kopírování konfiguračního souboru ze síťového zařízení na server Eagle. Po změně IP adresy serveru Eagle na adresu patřící do sítě laboratorní úlohy dochází ke kopírování konfiguračního souboru ze síťového zařízení a server Eagle obdobně jak při konfiguraci síťového zařízení v kapitole 2.5.2. Tentokrát se ale kopírují soubory ze síťového zařízení na server Eagle a ne naopak. Po ukončení procesu kopírování se IP adresa serveru Eagle opět nastavuje zpět na *standardní*<sup>18</sup>.

Změna IP adresy a výchozí brány serveru Eagle je prováděna pomocí EXPECT skriptu *ipchange.exp* (viz kapitola 2.5.3).



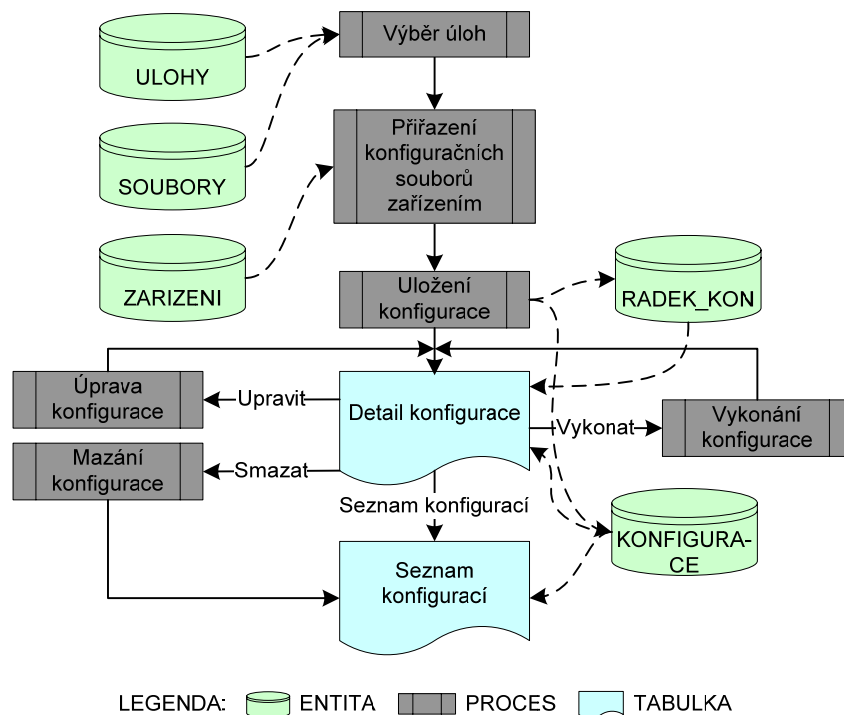
Obr. 11: Vývojový diagram nahrávání konfiguračního souboru ze síťového zařízení na server Eagle

## 2.4.7 Vytváření a vykonávání konfigurací

Úlohou skriptu *konfigurace.php* je vytváření a vykonávání konfigurací na síťových zařízeních. Obr. 12 demonstruje postup, jakým se ve skriptu *konfigurace.php* vytvářejí nové konfigurace, které je možné následně vykonat, případně editovat. Vytváření nových konfigurací v systému *AutoConf* je proces, při kterém se spojují do dvojic identifikátor

<sup>18</sup>Přednastavená IP adresa privátní lokální sítě LAN serveru Eagle.

síťového zařízení (*ZARIZENI.ID*) a identifikátor konfiguračního souboru (*SOUBORY.ID*). Takto vytvořené dvojice přesně definují, jaký konfigurační soubor má být nahrán jakému síťovému zařízení. Vytvořená dvojice identifikátorů spolu s identifikátorem konfigurace (*KONFIGURACE.ID*), do které daná dvojice patří, tvoří jeden řádek konfigurace a je uložena v entitě *RADEK\_KON*. Jména a celkové stavy konfigurací jsou uloženy v entitě *KONFIGURACE*. Každá konfigurace obsahuje alespoň jeden záznam v entitě *RADEK\_KON*. Při zobrazení detailu vytvořené konfigurace (viz Obr. 13) se automaticky spouští EXPECT skript *tester.exp*, a to pro každé zařízení použité v dané konfiguraci. Výsledek skriptu *tester.exp*, zobrazený ve sloupci *Stav*, má pouze informační charakter. Informuje uživatele, zda je síťové zařízení momentálně připojeno nebo odpojeno. Pomocí zaškrtnutých políček si uživatel může vybrat, která zařízení se chce v dané konfiguraci pokusit nakonfigurovat.



**Obr. 12: Postup vytváření konfigurace v systému pro automatickou konfiguraci**

Pro vykonání konfigurace, to znamená nahrání konfiguračního souboru do běžící konfigurace síťového zařízení, se volá EXPECT skript *configurator.exp*. Konfigurovat síťové zařízení je možné pouze v okamžiku, kdy je síťové zařízení připojeno. Proto se skript *configurator.exp* nejdříve pokouší k danému zařízení připojit a potom vykonat danou konfiguraci. Pokud se mu to nepodaří, vrátí výsledek chybu, která nastala při konfiguraci (viz seznam stavů v Tab. 3).

DETAIL KONFIGURACE						
KONFIGURACE č. 1						
	Typ	Název	IP adresa	Úloha	Konfigurační soubor	Stav
<input type="checkbox"/>	Router	Rack1-R11	192.168.254.11	Chapter 2 Lab 2-1, EIGRP Configuration, Bandwidth, and Adjacencies	R1	Odpojeno!
<input type="checkbox"/>	Router	Rack1-R12	192.168.254.12	Chapter 2 Lab 2-1, EIGRP Configuration, Bandwidth, and Adjacencies	R2	Odpojeno!
<input type="checkbox"/>	Switch	Rack1-S21	192.168.254.51	Chapter 2 Lab 2-1, EIGRP Configuration, Bandwidth, and Adjacencies	SW1	Odpojeno!

Obr. 13: Ukázka konfigurace vytvořené v systému pro automatickou konfiguraci

#### 2.4.8 Seznam konfigurací uložených v databázi systému

Správu nad všemi konfiguracemi v systému *AutoConf* má na starost modul *seznam\_kon.php*. Uživatelé poskytují informace o existujících konfiguracích uložených v databázi a zpřístupňuje další moduly pro vytváření a vykonávání konfigurací.

Obr. 14 je ukázkou seznamu existujících konfigurací. Sloupec *Úspěšnost* zde definuje operaci, která byla s danou konfigurací naposled udělána. Konfigurace zde může být *nová* nebo *změněná*, nebo *vykonána*. V tom případě sloupec *Úspěšnost* značí procento celkové úspěšnosti dané konfigurace. Přesněji, kolik síťových zařízení se v dané konfiguraci podařilo správně nakonfigurovat.

SEZNAM EXISTUJÍCÍCH KONFIGURACÍ				
Název konfigurace	Datum uložení	Majitel	Úspěšnost	Možnosti
KONFIGURACE č. 1	2010-12-07 14:35:01	Ucitel	Nová	<input type="button" value="Prohlédnout"/>
OSPF	2010-11-03 17:51:46	Ucitel	100 %	<input type="button" value="Prohlédnout"/>
Prázdná konfigurace na všechny	2010-12-07 14:45:16	Ucitel	0 %	<input type="button" value="Prohlédnout"/>

Obr. 14: Ukázka seznamu existujících konfigurací

#### 2.4.9 Modul pro rychlou konfiguraci síťového zařízení

Rychlá konfigurace (viz Obr. 15) je speciální funkce, kterou systém *AutoConf* nabízí. Představuje situaci, kdy uživatel může vykonávat konfiguraci síťových zařízení bez zdlouhavého postupu jejího vytváření. Uživatel při ní pouze přiřadí síťovému zařízení libovolný konfigurační soubor ze všech úloh a konfiguraci hned vykoná. Do databáze se při tom neukládá žádný záznam o dané konfiguraci. Možnost vykonávat rychlou konfiguraci je v mém základním nastavení povolena pouze u uživatele *Ucitel*.

RYCHLÁ KONFIGURACE					
Typ	Název	IP adresa	Stav	Konfigurační soubor	Možnosti
Router	Rack1-R11	192.168.254.11	Odpojeno!	--Vyberte konfigurační soubor!--	Konfiguruj
Router	Rack1-R12	192.168.254.12	Odpojeno!	--Vyberte konfigurační soubor!--	Konfiguruj
Router	Rack1-R13	192.168.254.13	Odpojeno!	--Vyberte konfigurační soubor!--	Konfiguruj
Router	Rack1-R14	192.168.254.14	Odpojeno!	Chapter 1 Lab 1-1, Tcl Script Referen... R1 R2	Konfiguruj
Router	Rack1-R15	192.168.254.15	Odpojeno!	Chapter 2 Lab 2-1, EIGRP Configurat... R1 R2	Konfiguruj
Router	Rack1-R16	192.168.254.16	Odpojeno!	R1 R2 R3	Konfiguruj
Switch	Rack1-S24	192.168.254.54	Odpojeno!		Konfiguruj

Obr. 15: Rychlá konfigurace síťových zařízení

## 2.5 Skripty pro terminál serveru Eagle

V systému *AutoConf* je několik EXPECT skriptů, z nichž každý má jinou úlohu:

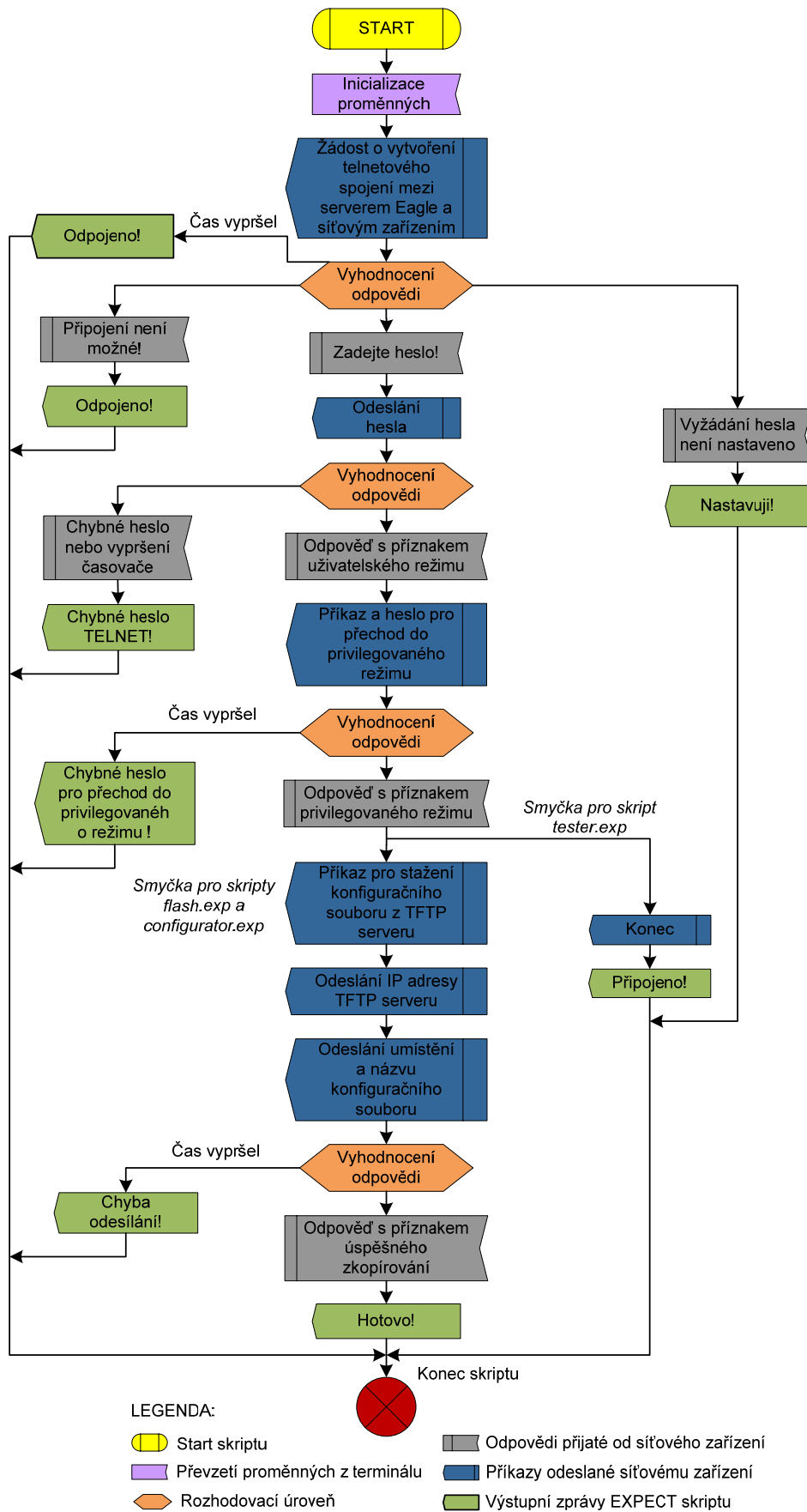
- *tester.exp* – testuje spojení serveru Eagle a síťového zařízení.
- *flash.exp* – nahrává TCL skript do paměti FLASH síťového zařízení.
- *configurator.exp* – do síťových zařízení kopíruje startovací a běžící konfigurace.
- *getconfig.exp* – určen ke kopírování konfiguračních souborů ze síťového zařízení na server.
- *ipchange.exp* – na serveru nastavuje IP adresu a výchozí bránu síťového rozhraní.
- *accessright.exp* – mění přístupová práva konfiguračních souborů na serveru.

### 2.5.1 Komunikace serveru Eagle se síťovým zařízením pomocí EXPECT skriptu

Pro komunikaci serveru Eagle se síťovými zařízeními se používají EXPECT skripty. Každý EXPECT skript nejdřív pomocí příkazu *spawn* vytvoří komunikační kanál TELNET mezi serverem Eagle a síťovým zařízením. Následně na terminál síťového zařízení odesílá textové příkazy příkazem *send* a příkazem *expect* vyhodnocuje příchozí hlášení zaslané síťovým zařízením na terminál. Při ladění běhu EXPECT skriptů jsem zjistil, že není potřeba, aby příkaz *expect* čekal na odpověď z terminálu déle než 5 vteřin. Proto jsem v každém skriptu nastavil proměnnou *timeout* na 5 vteřin.

### 2.5.2 Postup konfigurace síťového zařízení pomocí EXPECT skriptu

Obr. 16 je vývojový diagram EXPECT skriptů *configurator.exp*, *flash.exp* a *tester.exp*. Všechny tyto skripty mají skoro stejný zdrojový kód. Kvůli přehlednosti systému byly ale rozděleny do více souborů. Skripty používají stejný způsob připojení k síťovému zařízení,



Obr. 16: Vývojový diagram EXPECT skriptu pro konfiguraci síťového zařízení

však *tester.exp* se hned po navázání spojení odpojí a skončí, *configurator.exp* pokračuje v konfigurování běžící konfigurace a *flash.exp* kopíruje TCL skript do paměti FLASH síťového zařízení.

### Postup běhu EXPECT skriptu:

- Inicializace proměnných – naplnění všech proměnných vstupními parametry EXPECT skriptu, nastavení proměnné *timeout* na 5 vteřin.
- Odeslání požadavku na vytvoření TELNET spojení se síťovým zařízením příkazem *spawn telnet \$IP*. V proměnné *\$IP* je uložena IP adresa síťového zařízení.
- Čekání na odpověď se žádostí hesla pro autorizaci TELNET spojení „*password*”.
- Pokud síťové zařízení odpoví hlášením „*Unable to connect*“, případně nepříjde odpověď do uplynutí času v proměnné *timeout*, skript skončí s hlášením: „*Odpojeno!*“
- Pokud odpoví hlášením „*Password required, but none set*“ znamená to, že síťové zařízení má povolenou komunikaci pomocí protokolu TELNET, ale není nastaveno heslo pro připojení. K této situaci dochází při inicializaci síťového zařízení. Skript v tomto případě odpoví hlášením: „*Nastavuji!*“
- V případě odpovědi „*password*“ skript pokračuje odesláním hesla pro autorizaci TELNET spojení, které je obsahem proměnné *\$TELNET*.
- Čekání na odpověď obsahující příznak přihlášení do uživatelského režimu „*>*“.
- Pokud síťové zařízení neodešle odpověď obsahující „*>*“ do uplynutí času v proměnné *timeout*, skript skončí s hlášením „*Chybné heslo TELNET!*“
- V případě odpovědi „*>*“, odeslání příkazu *enable* pro přechod do privilegovaného režimu síťového zařízení.
- Odeslání hesla pro přechod do privilegovaného režimu v proměnné *\$PRIVILEG*.
- Čekání na odpověď obsahující příznak přihlášení do uživatelského režimu „*#*“
- Pokud síťové zařízení neodešle do uplynutí času v proměnné *timeout* nebo je odpovědí zpráva „*Bad password*“, skript skončí s hlášením „*Chybné heslo #!*“
- Pokud proces vrátí hlášení obsahující příznak „*#*“ znamená to, že spojení se síťovým zařízením proběhlo úspěšně. EXPECT skript *tester.exp* proto odešle příkaz *exit* pro ukončení spojení a skončí s hlášením „*Připojeno!*“
- EXPECT skripty *configurator.exp* a *flash.exp* pokračují v konfiguraci odesláním příkazu „*copy tftp \$PRIKAZ*“, kde proměnná *\$PRIKAZ* obsahuje text „*running-config*“ pro nahrávání konfiguračních souborů ze serveru Eagle na síťové zařízení ve skriptu

*configurator.exp*, nebo „*flash*“ pro nahrání TCL skriptu do paměti FLASH v síťovém zařízení skriptem *flash.php*.

- Dále jsou odeslány parametry pro kopírování souboru. *\$ZDROJ* obsahuje IP adresu serveru Eagle jako zdroje souboru pro kopírování. Parametr v proměnných *\$UMISTENI\$SOUBOR* označuje soubor a jeho úložiště na serveru Eagle.
- Skript *flash.exp* odesílá ještě informaci o tom, pod jakým názvem se má TCL skript uložit v paměti FLASH, aby jej EEM applet mohl spustit.
- Poté v síťovém zařízení probíhá pokus o nahrání požadovaného souboru.
- EXPECT skript čeká na odpověď obsahující příznak „OK“, který značí úspěšnost nahrání souboru ze serveru Eagle na síťové zařízení.
- Pokud síťové zařízení neodpoví odpovědí obsahující příznak „OK“, EXPECT skript skončí s hlášením „*Chyba odesílání!*“.
- Pokud proces odpoví odpovědí obsahující příznak „OK“ znamená to, že konfigurační soubor se podařilo nahrát na síťové zařízení. EXPECT skript potom skončí s hlášením „*Hotovo!*“.

V případě, že na síťové zařízení nahráváme novou běžící konfiguraci, síťové zařízení po nahrání konfiguračního souboru odpoví procesu hlášením obsahujícím příznak „OK“ a hned potom spustí novou běžící konfiguraci. To způsobí softwarové odpojení síťového zařízení ze systému *AutoConf*, protože nová konfigurace má pravděpodobně jinak nastavenou konfiguraci než požaduje systém *AutoConf*.

### 2.5.3 Skript pro změnu IP adresy a výchozí brány serveru Eagle

Proces změny IP adresy serveru Eagle je naznačen na Obr. 17 Během tohoto procesu se PHP engine připojuje na BASH rozhraní serveru Eagle pomocí protokolu SSH. Poté se pomocí příkazu

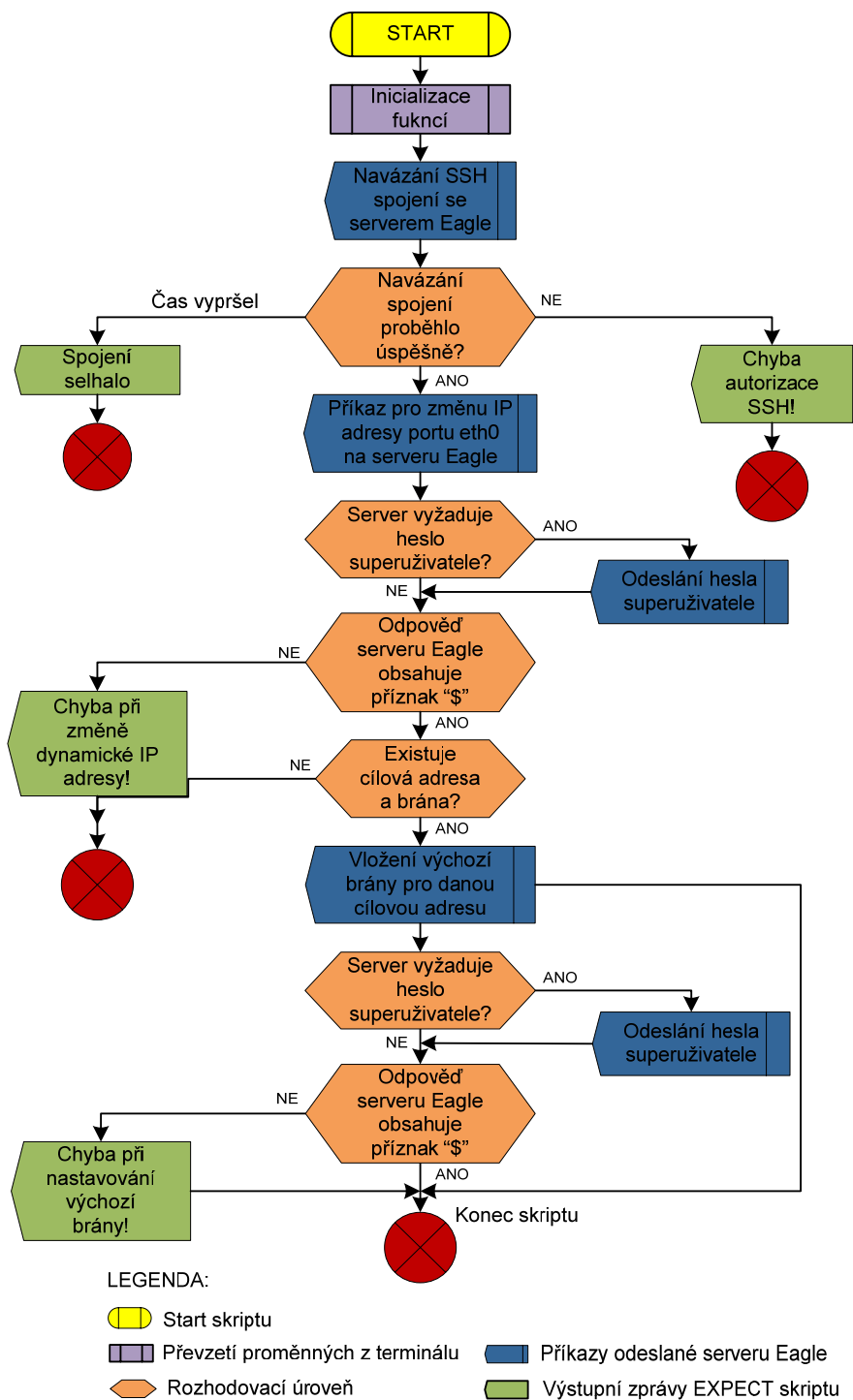
```
sudo /sbin/ifconfig $PORT $PORTIP netmask $MASKA up
```

pokouší změnit IP adresu privátního ethernetového rozhraní serveru Eagle. Kde parametry *\$PORT*, *\$PORTIP* a *\$MASKA* vyjadřují název, IP adresu a masku konfigurovaného síťového rozhraní. Dále pak pomocí příkazu

```
sudo /sbin/route add -host $DESTINATION gw $GATEWAY dev $PORT
```

nastavuje výchozí bránu (parametr *\$GATEWAY*) pro data směřované na síťové rozhraní (*\$DESTINATION*) zařízení, ze kterého budeme kopírovat konfigurační soubor.

Proces změny IP adresy a nastavení výchozí brány je na serveru Eagle vykonáván pomocí skriptu *ipchange.exp*. Jedním ze vstupních parametrů skriptu *ipchange.exp* je heslo pro uživatele *sudo*<sup>19</sup>, které umožňuje konfigurovat server Eagle jako superuživatel<sup>20</sup>.



Obr. 17: Změna IP adresy serveru Eagle

<sup>19</sup>Heslo pro vykonání operace s oprávněními jiného uživatele, jímž je obvykle root (Superuser).

<sup>20</sup>Superuživatel (root) má v operačním systému linux roly systémového administrátora s nejvyšším oprávněním.

## 2.6 Instalace systému pro automatickou konfiguraci

### 2.6.1 Potřebné softwarové nástroje

Pro instalaci systému *Autoconf* na server Eagle je nutné, aby na něm pracoval operační systém Linux. Dále musíme nainstalovat a správně nastavit následující softwarové nástroje:

- LAMP server – balík, který obsahuje programy potřebné pro činnost webových stránek:
  - Apache http server – webový server. Na adresář systému *AutoConf* je potřebné nastavit minimálně jméno a kořenový adresář HTTP serveru.
  - MySQL – relační databáze. Systém byl navržen a testován v databázi MySQL verze 5.
  - PHP – skriptovací jazyk pro dynamizaci webových stránek. Systém byl navržen pro použití PHP verze 5.
- Balíček SSH2 – tento program umožňuje serveru PHP komunikaci s terminálem serveru Eagle pomocí zabezpečeného spojení SSH.
- Balíček EXPECT – pomocí něj je možné spouštět EXPECT skripty v příkazovém řádku serveru Eagle.
- TFTP server – umožňuje kopírování konfiguračních souborů ze serveru Eagle na síťové zařízení. Je potřeba nastavit TFTP server pro možnost zadávání úplné adresy konfiguračního souboru, určeného pro kopírování.
- DHCP server – pomocí něj se zajišťuje přidělování IP adres pro síťová zařízení při jejich inicializaci do systému *AutoConf*.

### 2.6.2 Nastavení systému

Pro nastavení databáze MySQL je potřeba vytvořit databázi s názvem *Autoconf* a definovat v ní všechny potřebné entity a relace. Soubor *Autoconf.sql* obsahuje zdrojový kód, který vytvoří databázi *Autoconf* a vloží do ní základní informace pro správnou funkci systému *AutoConf*, například typy síťových zařízení, použité v systému *AutoConf*, a informace o uživateli a uživatelských právech. Pro správnou funkci všech funkcí musí být databáze datového typu *InnoDB*.

Adresář s názvem *autoconf*, který obsahuje všechny skripty systému *AutoConf*, je potřeba zkopírovat do adresáře */var/www/* paměti serveru Eagle.

Pro umožnění komunikace PHP funkcí s databází MySQL a s terminálem serveru Eagle je nutné editovat proměnné s přihlašovacími údaji ve skriptech *db\_spojeni.php* a *ssh\_spojeni.php*. Základní funkce v těchto skriptech jsou popsány v kapitolách 2.4.1 a 2.4.2.

Funkce *nastaveni()* je uložena ve skriptu *ssh\_spojeni.php*. Tato funkce vrací pole proměnných, obsahující základní parametry využívané při činnosti systému *AutoConf*. Při instalaci systému *AutoConf* je nutné správně nastavit hodnoty proměnných, uložené v této funkci. Význam jednotlivých proměnných je popsán v následujícím zdrojovém kódu.

```
function nastaveni() {
// IP adresa pro přihlašování na BASH rozhraní serveru Eagle.
$konstanta["ssh2ip"] = "127.0.0.1";
$konstanta["ssh2login"] = "Login superuživatele";
$konstanta["ssh2pass"] = "Heslo superuživatele";
// Rozhraní, kterému se bude dynamicky měnit IP adresa.
$konstanta["rozhrani"] = "eth0";
// IP adresa a maska serveru Eagle. Využívá se při kopírování
// konfiguračních souborů ze serveru Eagle na síťové zařízení a
// naopak.
$konstanta["zdroj"] = "193.168.10.100";
$konstanta["zdrojmask"] = "255.255.255.0";
// Úplná adresa adresáře, ve kterém se nacházejí konfigurační
// soubory pro síťová zařízení.
$konstanta["adresar_souboru"]="umístění adresáře soubory/";
// Adresa adresáře, ve kterém se nacházejí konfigurační soubory pro
// síťová zařízení využívaná při komunikaci pomocí TFTP.
$konstanta["adresar_souboru_tftp"]="";
// Úplná adresa adresáře, ve kterém se nacházejí skripty pro
// komunikaci serveru Eagle se síťovými zařízeními.
$konstanta["adresar_bash"] = "umístění adresáře bash/";
return $konstanta;
}
```

### 1.1.1 Softwarové nástroje a verze použité při testování

- Ubuntu 8.10 – distribuce operačního systému Linux, použitá na serveru Eagle.
- Apache 2.2.9 (Ubuntu) – webový server.
- MySQL 5.0.67-0ubuntu6 – relační databáze.
- PHP 5.2.6-2ubuntu4 – skriptovací jazyk pro dynamizaci stránek.
- prototype.js 1.6.0.3 – knihovna funkcí AJAX.
- Screem 0.16.1 – editor pro psaní zdrojového kódu webových stránek.
- Firefox 3.0.10, Internet Explorer 6.0 – prohlížeč webových stránek.

### 3 Závěr

V této diplomové práci byl navržen systém s názvem *AutoConf* pro automatickou konfiguraci síťových prvků v laboratoři Cisco akademie. Tento systém byl vytvořen za účelem usnadnění přípravy laboratorních úloh vykonávaných na přepínačích a směrovačích.

Systém je možné ovládat pomocí webových stránek, umístěných na serveru Eagle. Uživatel může do systému vkládat nové laboratorní úlohy a pak jejich konfigurační soubory přidělovat síťovým zařízením, připojeným do systému *AutoConf*. Následně je pak možná jejich automatická konfigurace. Informace o síťových zařízeních, laboratorních úlohách, konfiguračních souborech, konfiguracích vytvořených uživatelem a uživatelských právech jsou uloženy v databázi MySQL.

Na směrovače, které pracují se systémem *AutoConf*, byl implementován mechanismus rozhodování, který zjišťuje, zda si uživatel přeje konfigurovat daný směrovač pomocí systému *AutoConf*. Pokud ano, je třeba, aby síťové zařízení před jeho zapnutím uživatel připojil k přepínači, na který je připojen server Eagle. Mechanismus pak toto spojení detekuje a nastaví vše potřebné pro komunikaci daného směrovače se serverem Eagle. V opačném případě se na směrovači nastaví základní konfigurace. Mechanismus rozhodování je však možné aplikovat pouze na směrovače a přepínače podporující EEM applety.

Konfigurační soubory síťových zařízení lze na server Eagle nahrávat pomocí webových stránek buď z počítače a nebo přímo ze síťových zařízení. Pro nahrávání konfiguračních souborů přímo ze síťového zařízení stačí server Eagle připojit do sítě síťového zařízení a správně vyplnit formulář pro nahrávání. Server Eagle pak změní svou IP adresu, připojí se na dané síťové zařízení a vyžádá si od něj konfigurační soubor, který je pak uložen do databáze systému *AutoConf*.

V rámci této diplomové práce byl také vytvořen systém uživatelů a uživatelských práv, pomocí kterého je zabezpečen přístup do webové aplikace systému *AutoConf*. Uživatelé se v něm rozlišují na učitele a studenty. Ti mají pak rozdílná uživatelská práva.

Funkční verze systému *AutoConf* je nainstalována na serveru v učebně PA-128 Fakulty elektrotechniky a komunikačních technologií Vysokého učení technického v Brně. Z internetu je možné na ní přistupovat na adrese <http://grid1.utko.feec.vutbr.cz/>.

## Použitá literatura

- [1] HICKS, Michael. *Cisco : Optimalizace aplikací*. Martin Holub; Petra Tesárková. Praha : Grada Publishing, a.s., 2008. 336 s. ISBN 978-80-247-1610-7.
- [2] BIGELOW, Stephen J. *Mistrovství v počítačových sítích : Správa, konfigurace, diagnostika a řešení problémů*. Petr Matějů. Brno : Computer Press, 2004. 990 s. ISBN 80-251-0178-9.
- [3] *Linux : Dokumentační projekt*. Přeložili Lubomír Ptáček, Jakub Mikulaščík, Veronika Matějů, Antonín Novotný, Jiří Veselský, Ondřej Zloský. 4. aktualiz. vyd. Brno : Computer Press, a.s., 2007. 1334 s. ISBN 978-80-251-1525-1.
- [4] HUCABY, David, MCQUERY, Steve. *Konfigurace směrovačů Cisco : Autorizovaný výukový průvodce*. Přeložil Jiří Veselský. Brno : Computer Press, a.s., 2004. 632 s. ISBN 80-722-6951-8.
- [5] MILAR, Bohdan. *Seriál o BASHi* [online]. 2005 [cit. 2008-12-15]. Dostupný z WWW: <<http://www.linuxexpres.cz/praxe/serial-o-bashi>>.
- [6] *Seriál TCL základy* [online]. 2004 [cit. 2011-05-18]. Dostupný z WWW: <<http://www.root.cz/serialy/tcl-zaklady/>>.
- [7] OPPEL, Andy. *SQL bez předchozích znalostí : průvodce pro samouky*. Redaktor Radek Hymlar; překlad Jan Gregor, Jakub Mikulaščík. 1. autoriz. vyd. Brno : Computer press, 2008. 240 s. ISBN 978-80-251-1707-1.
- [8] GILMORE, W. Jason. *Velká kniha PHP5 & MySQL : Kompendium znalostí pro začátečníky i profesionály*. Miroslav Kučera; RNDr. Jan Pokorný. 1. vyd. Brno : Zoner Press, 2005. 711 s. ISBN 80-86815-20-X.
- [9] NARAMORE, Elizabeth, et al. *PHP5, MySQL, Apache : Vytváříme webové aplikace*. Sazba Bogdan Kiszka; odpovědný redaktor Martin Domes; přeložil Bogdan Kiszka. 1. vyd. Brno : Computer Press, a.s., 2006. 813 s. ISBN 80-251-1073-7.
- [10] KOFLER, Michael . *Mistrovství v MySQL 5 : Kompletní průvodce webového vývojáře*. 1. vyd. Brno : Computer press, 2007. 808 s. ISBN 978-80-251-1502-2.
- [11] OPPEL, Andrew. *Databáze bez předchozích znalostí : průvodce pro samouky*. Přeložil David Krásenský. 1. vyd. Brno : Computer Press, 2006. 319 s. Obsahuje rejstřík. ISBN 80-251-1199-7.
- [12] MACH, Jakub. *PHP pro úplné začátečníky* . 2. rozš. vyd. Brno : Computer Press, 2006. 167 s. ISBN 80-251-1248-9.

- [13] VELTE, Toby J., VELTE, Anthony T. *Síťové tefnologie CISCO : Velký průvodce*.
- [14] David Krásenský. Brno : Computer Press, 2003. ISBN 80-7226-857-0. s. 0-759. Cisco akademie, *UTP kabeláž v laboratoři* [online]. c2008 [cit. 2011.05.18]. Dostupné z WWW:<[http://adela.utko.feec.vutbr.cz/cisco\\_akademie/index.php?option=com\\_content&task=view&id=17&Itemid=33](http://adela.utko.feec.vutbr.cz/cisco_akademie/index.php?option=com_content&task=view&id=17&Itemid=33)>.
- [15] STEPHENSON, Sam. *Prototype JavaScript framework* [online]. 2006-2007 [cit. 2011-05-18]. Stránky pro stažení knihovny. Dostupný z WWW:<<http://www.prototypejs.org/download>>.

## Seznam zkratek

UTP	– Unshielded Twisted Pair
IP	– Internet Protocol
OSI	– Open System Interconnection
ISO	– International Standardization Organization
ISDN	– Integrated Services Digital Network
RAM	– <i>Random-Access Memory</i>
NVRAM	– Nonvolatile RAM
ROM	– Read-Only Memory
IOS	– Internetwork Operating System
AUX	– Auxiliary port
WAN	– Wide Area Network
MAC	– Media Access Control
VLAN	– Virtual Local Area Network
TCL	– Tool Command Language
EEM	– Embedded Event Manager
TELNET	– Telecommunication Network
CLI	– Command Line Interpreter
TFTP	– Trivial File Transfer Protocol
TCLSH	– Tool Command Language shell
BASH	– Bourne Again Shell
FTP	– File Transfer Protocol
SSHD	– Secure Shell Daemon
RSA	– Rivest Shamir Adelman
LAMP	– Linux Apache MySQL PHP server
HTML	– HyperText Markup Language
CSS	– Cascading Style Sheets
AJAX	– Asynchronous JavaScript and XML
DHCP	– Dynamic Host Configuration Protocol
ARP	– Address Resolution Protocol

## Seznam příloh

A	Stromová struktura adresáře <i>System_AutoConf</i> .....	60
B	Obsah přiloženého CD.....	61

## **A Stromová struktura adresáře *autoconf***

### ***ajax\_lib/***

*prototype.js*  
*readme.txt*

### ***autentizace/***

*autentizace.php*  
*kontrola.php*

### ***bash/***

*accessright.exp*  
*configurator.exp*  
*flash.exp*  
*getConfig.exp*  
*ipchange.exp*  
*tester.exp*

### ***db\_skripty/***

*db\_funkce.php*  
*db\_spojeni.php*

### ***navody/***

#### ***phpkbash/***

*bash.php*  
*bashrychle.php*  
*device.php*  
*flash.php*  
*inicializace.php*  
*ssh\_spojeni.php*

### ***soubory/***

#### ***flash/***

*autoconf\_rack1-r1*  
*network-config*  
*error.log*

### ***uvod/***

#### ***vzhled/***

*hlavicka.php*  
*nabidka\_vlevo.php*  
*styly.css*  
*zadny.gif*  
***obrazky/***

#### *form.js*

*konfigurace.php*  
*php\_funkce.php*  
*prihlasit.php*  
*rychla\_kon.php*  
*seznam\_konf.php*  
*ulohy.php*  
*uvod.php*  
*zarizeni.php*  
*zobrazobr.php*

## B Obsah přiloženého CD

Diplomova_prace.pdf	– diplomová práce
Autoconf.sql	– zdrojový kód databáze MySQL
autoconf	– adresář obsahující zdrojové kódy systému <i>AutoConf</i>