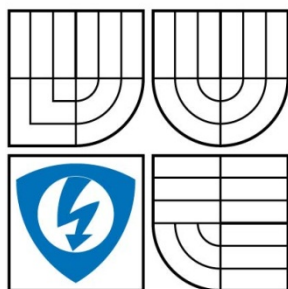


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



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

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LUKÁŠ VERNER

VEDOUCÍ PRÁCE
SUPERVISOR

ING. DAN KOMOSNÝ, PH.D.

BRNO 2009

Originální zadání...

Anotace

Tato bakalářská práce se zabývá návrhem a implementací systému pro automatickou konfiguraci síťových zařízení Cisco. Systém se jmenuje *AutoConf* a je určen pro vzdálené kopírování konfiguračních souborů z eagle-serveru na přepínače a směrovače v učebně Cisco akademie. Ovládán je pomocí webových stránek umístěných na eagle-serveru.

V tomto systému smí uživatel vytvářet nové laboratorní cvičení s konfiguračními soubory pro síťová zařízení, a pak je použít v konfiguracích. Informace o vykonaných konfiguracích jsou umístěny v relační databázi MySQL.

Síťová zařízení, která budou konfigurována pomocí systému *AutoConf*, musí být spojena s eagle-serverem. Pro nastavení spojení mezi síťovým zařízením a eagle-serverem byl vytvořen mechanismus rozhodování. Pokud při startu síťového zařízení mechanismus detekuje fyzické spojení síťového zařízení s eagle-serverem, nastaví dané zařízení pro komunikaci s eagle-server.

Klíčová slova

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

Abstract

This bachelor'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 distant copying configuration files from eagle-server to the switches and routers in Cisco academy laboratory. It is handled by web pages placed on eagle-server.

In this system, user can create new laboratory exercises with configuration files for network devices and then these files could be used for configurations. Informations about done configurations are saved in relation database MySQL.

Network devices which will be configured by system *AutoConf* have to be connected with eagle-server. To setup their connection settings was created decision-making mechanism. If it detects physical connection between network device and eagle-server before network device starts, it will set network device for communication with eagle-server.

Key words

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

VERNER, L. *Automatická konfigurace síťových prvků Cisco akademie*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 61 s. Vedoucí bakalářské práce Ing. Dan Komosný, Ph.D.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma AUTOMATICKÁ KONFIGURACE SÍŤOVÝCH PRVKŮ CISCO AKADEMIE jsem vypracoval samostatně pod vedením vedoucího bakalářské práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské 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 bakalářské práce Ing. Danu Komosnému, Ph.D. za velmi užitečnou metodickou pomoc a cenné rady při zpracování práce.

V Brně dne

.....

podpis autora

Obsah

Úvod.....	9
1 Teoretický úvod.....	11
1.1 Popis učebny Cisco akademie	11
1.1.1 Směrovače	12
1.1.2 Přepínače	14
1.2 Konfigurace síťových zařízení s operačním systémem IOS.....	15
1.2.1 Operační systém Cisco IOS.....	16
1.2.2 Základní příkazy IOS	17
1.3 Skriptovací jazyk TCL v operačním systému IOS	17
1.4 Detekce událostí v Cisco IOS s využitím nástroje EEM	18
1.5 Operační systém CentOS.....	19
1.6 Vykonavatel Linuxových textových příkazů BASH	19
1.7 Nástroj pro automatizaci interaktivních aplikací EXPECT.....	20
1.8 Vzdálená komunikace mezi klientem a síťovým zařízením.....	21
1.8.1 Komunikační protokol TELNET	21
1.8.2 Bezpečná konfigurace síťových zařízení pomocí SSH	21
1.9 Sada programů platformy LAMP	21
1.9.1 Webový server Apache	22
1.9.2 Relační databáze MySQL.....	22
1.9.3 Programovací jazyk PHP	26
2 Automatická konfigurace síťových prvků Cisco Akademie	28
2.1 Základní funkce systému pro automatickou konfiguraci síťových zařízení Cisco....	28
2.2 Inicializace síťového zařízení v systému <i>AutoConf</i>	30
2.2.1 Detekce zapnutí směrovače pomocí EEM appletu.....	30
2.2.2 TCL skript pro konfiguraci směrovače	31
2.2.3 Inicializace přepínače v systému <i>Autoconf</i>	35
2.3 <i>Autoconf</i> – databáze systému pro automatickou konfiguraci.....	35
2.4 Webové moduly systému <i>AutoConf</i>	40
2.4.1 Funkce pro připojení PHP do databáze MySQL	41
2.4.2 Funkce pro SSH spojení vykonavatele PHP příkazů s interpretem BASH.....	41
2.4.3 Autorizace uživatelů do systému <i>AutoConf</i>	42

2.4.4	Seznam zařízení pro automatickou konfiguraci	43
2.4.5	Seznam laboratorních úloh a konfiguračních souborů	44
2.4.6	Vytváření a vykonávání konfigurací v systému <i>AutoConf</i>	46
2.4.7	Seznam konfigurací uložených v databázi systému <i>AutoConf</i>	48
2.4.8	Modul pro rychlou konfiguraci síťového zařízení	48
2.5	Skripty pro terminál eagle-serveru	49
2.5.1	Komunikace eagle-serveru se síťovým zařízením pomocí EXPECT skriptu	49
2.5.2	Příkazy BASH, volané pomocí PHP skriptu systému <i>AutoConf</i>	52
2.6	Instalace systému <i>AutoConf</i>	53
2.6.1	Potřebné softwarové nástroje	53
2.6.2	Nastavení systému <i>AutoConf</i>	53
2.6.3	Softwarové nástroje, použité při testování	54
3	Závěr.....	55
	Použitá literatura	56
	Seznam zkratek	58
	Seznam příloh.....	59

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 konfigurovat všechna síťová zařízení z jednoho počítače najednou.

V této bakalářské práci jsem vytvořil komplexní systém pro automatickou konfiguraci síťových zařízení. Automatická konfigurace je zaměřena pouze na přepínače a směrovače nacházející se v učebně PA-128 a je ovládána pomocí sady skriptů umístěných na specifickém počítači v učebně, eagle-serveru. Činnost skriptu je ovládána pomocí webové aplikace rovněž umístěné na eagle-serveru. 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 prázdný 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 s eagle-serverem. V případě, že bychom je nechtěli vzdáleně konfigurovat, by to bylo zbytečné. Za tímto účelem jsem se pokusil vytvořit mechanismus rozhodování, založený na spojení s eagle-serverem. Pokud chce uživatel zařízení konfigurovat z eagle-serveru, stačí, když jej před zapnutím připojí k přepínači eagle-serveru. Mechanismus rozhodování, který je v síťovém zařízení, toto připojení detekuje a nastaví vše potřebné, aby umožnil komunikaci eagle-serveru se síťovým zařízením. Jestliže ale mechanismus rozpozná, že zařízení při svém startu není připojeno k přepínači eagle-serveru, bude toto zařízení automaticky nastaveno s prázdnou běžící konfigurací.

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. Datovou strukturu webových stránek zde tvoří databáze umístěná na eagle-serveru. 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 eagle-serveru. 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 z eagle-serveru 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 eagle-serveru a zapojit jej do sítě podle schématu zapojení dané laboratorní úlohy.

Výsledkem tohoto projektu by měl být systém pro automatickou konfiguraci síťových zařízení Cisco, usnadňující učitelů přípravu učebny na vypracování laboratorních úloh na přepínačích a směrovačích. Měl by být ovládán pomocí webového rozhraní umístěného na eagle-serveru.

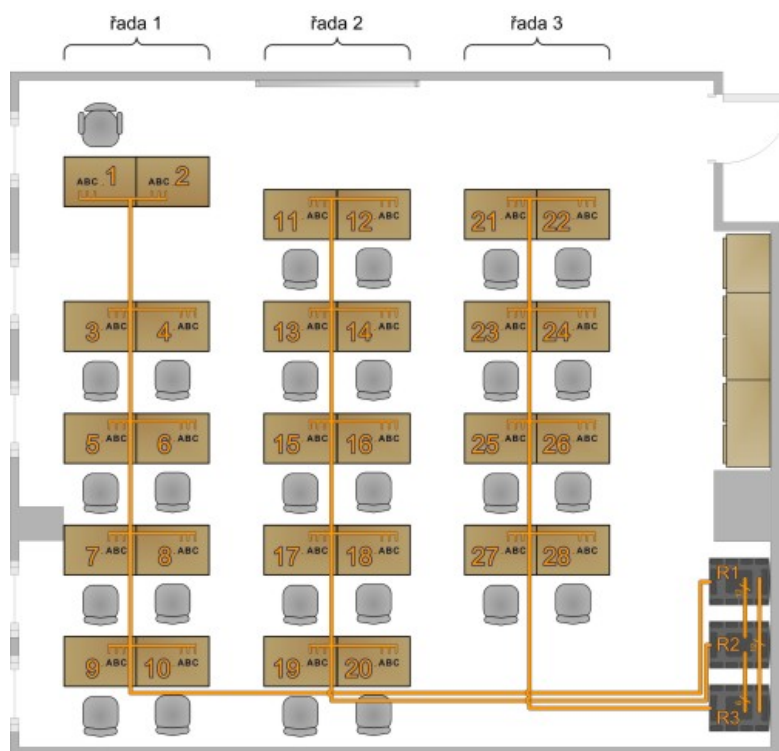
1 Teoretický úvod

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 eagle-server. 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, dvěmi síťovými kartami a jedním sériovým rozhraním. Pomocí UTP (Unshielded Twisted Pair) kabelů jsou tato rozhraní napojena k rozvaděčům v rohu místnosti (viz Obr. 1).

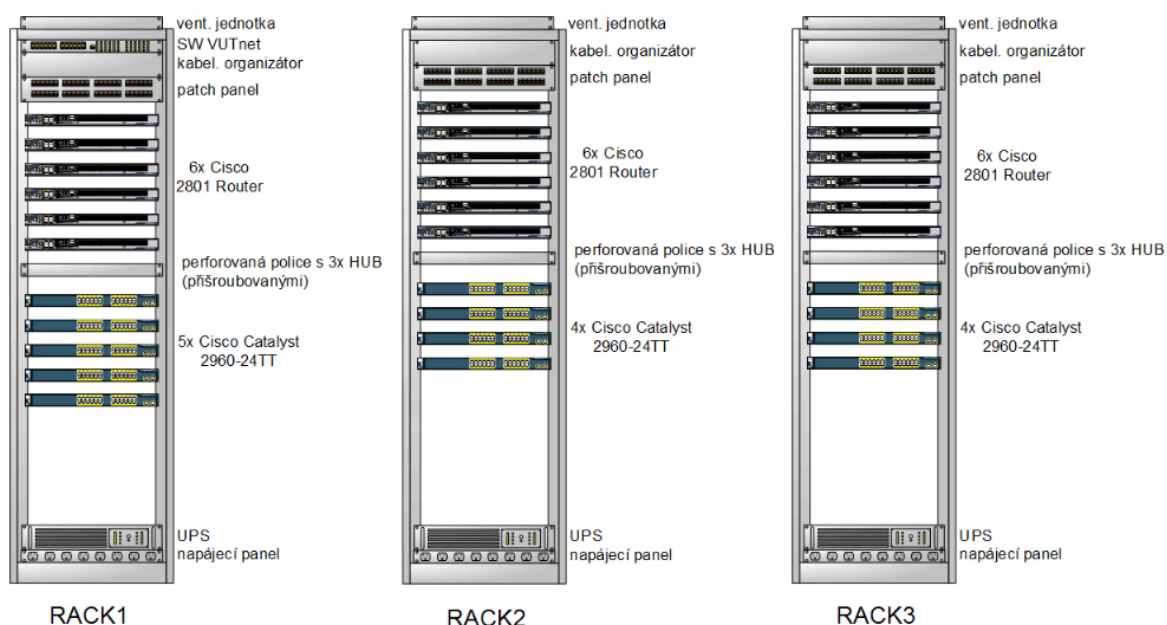


Obr. 1: Náskres učebny PA-128 – převzato z [15]

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. Sít'ová rozhraní počítačových stanic jsou v rozvaděčích připojena k *patch panelům* a označena podle čísla stolu počítačové stanice a typu sít'ového rozhraní (viz Obr. 2). Sít'ová rozhraní počítačových stanic jsou v patch panelech značeny A, B a C. Jako A je označena sít'ová karta, která umožňuje přístup do sítě VUT. Rozhraním se značkou B se míní sériové rozhraní, jehož se využívá pro konfiguraci sít'ových zařízení pomocí konzolového připojení s využitím kabelu RJ45-DB9. Zbývající sít'ové rozhraní, tj. druhá sít'ová 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 a 13 přepínačů série 2960-24TT. Tato aktivní sít'ová zařízení jsou podrobněji popsána níže [15].



Obr. 2: Rozmístění přepínačů a směrovačů v učebně PA-128– převzato z [15]

1.1.1 Směrovače

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

¹ Sled informací, na úrovni sít'ové vrstvy, který je přenášen určitou sítí.

² 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*³.
- 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*⁴,
- 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

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

⁴ Pomocí programu *bootstrap* lze systém směrovače zavést do podoby minimálního konfiguračního souboru.

je síťové rozhraní, které se zasouvá do skříně směrovače. Mezi nejčastější typy síťových rozhraní, které se ve směrovačích používají, patří ethernetová rozhraní, sériová synchronní rozhraní, bezdrátová rozhraní nebo rozhraní pro připojení do sítě ISDN [1], [2], [14].

Směrovač série Cisco 2800 umístěné v učebně PA-128

Směrovače série Cisco 2800 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*⁵ 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*⁶ (*Media Access Control*) adres připojených síťových zařízení. Přepínač pravidelně zkoumá, na jakém potru má připojené jaké síťové zařízení a zjištěné informace si ukládá do tabulky MAC adres. Následně je při přechodu *rámce*⁷ 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.

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

⁶ Hardwarová adresa síťové karty, případně síťového rozhraní.

⁷ 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*⁸ 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 sítě do virtuálních sítí VLAN (Virtual Local Area Network) a tím snadnější správa a management uživatelských skupin sítě [2].

Přepínače řady Catalyst 2960-24TT

Tyto přepínače se v učení 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 tohoto přepínače je, že nepodporuje konfiguraci pomocí skriptovacího jazyka TCL (Tool Command Language) ani nástroj EEM (Embedded Event Manager). To omezuje jeho funkci v systému automatické konfigurace, vytvořeném v této bakalářské práci.

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,

⁸ 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 víceúlohový 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 Základní příkazy IOS

V následující části jsou popsány základní příkazy pro konfiguraci přepínačů a směrovačů, využité v této bakalářské práci:

- **enable** – příkaz pro přechod z uživatelského módu do privilegovaného. Z bezpečnostních důvodů se přechod do privilegovaného módu zabezpečuje heslem.
- **configure terminal** – příkaz pro přechod z privilegovaného módu do konfiguračního módu.
- **hostname <jméno>** – nastavení jména zařízení.
- **interface <rozhraní> <číslo>** – příkaz pro přechod do režimu konfigurace síťového rozhraní. Například *FastEthernet0/0*, nebo *Serial0/1/0* při konfiguraci směrovače, *vlan 1* v přepínačích.
- **ip address <adresa> <maska>** – nastavení IP adresy a masky síťového rozhraní.
- **no shutdown** – aktivace síťového rozhraní.
- **exit** – přechod do podřazeného konfiguračního módu
- **line vty 0 4** – nastavení počtu možných současných spojení pomocí protokolu TELNET na 5 spojení. Ve směrovačích se nejčastěji nastavuje na 5 (*line vty 0 4*) a v přepínačích 16 (*line vty 0 15*).
- **password <heslo>** – nastavení hesla pro připojení přes TELNET.
- **login** – příkaz nutný pro umožnění kontroly, heslo při přihlašování do síťového zařízení. Pokud chceme použít přihlašovací jména, tak za příkaz napíšeme slovo *local* a v konfiguračním režimu definujeme uživatele a jejich přihlašovací hesla..
- **enable secret <heslo>** – nastavení bezpečnostního hesla pro přechod z uživatelského do privilegovaného módu.
- **copy <zdroj> <cíl>** – v privilegovaném režimu slouží ke kopírování souborů.
- **copy running-config startup-config** – uložení běžící konfigurace do startovací konfigurace v NVRAM.
- **copy tftp: running-config** – Nahrání běžící konfigurace z TFTP (Trivial File Transfer Protocol) serveru.

1.3 Skriptovací jazyk TCL v operačním systému IOS

IOS obsahuje TCL 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 interpreter 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.

Povolení a spuštění TCL v Cisco IOS

- a) **enable** – vstup do privilegovaného režimu.
- b) **configure terminal** – vstup do globálního konfiguračního režimu.
- c) **scripting tcl encdir <cesta>** – specifikace cílové cesty externích kódovacích souborů, které budou využívány TCL příkazem *encoding*.
- d) **scripting tcl int <cesta>** – nastavení inicializačního skriptu, který bude spuštěn po povolení TCL shellu.
- e) **tcsh** – povolení interaktivního TCL shellu a vstup do TCL konfiguračního módu, kde se zadávají TCL příkazy [7].

1.4 Detekce událostí v Cisco 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. Narozdíl od EEM appletu můžou 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.

Oznamování událostí protokolem SYSLOG

Protokol SYSLOG se používá v síťových zařízeních pro zveřejňování nevyžádaných zpráv správní stanici. Nevyžádané zprávy informují uživatele o akcích provedených systémem. Například o změnách stavu rozhraní nebo o systémovém restartu. To může být použito jako rozhodovací úroveň při konfiguraci zařízení [1].

1.5 Operační systém CentOS

CentOS patří mezi jednu z mnoha distribucí operačního systému Linux. Tento operační systém je nainstalován na pracovních stanicích v učebně Cisco akademie, včetně eagle-serveru.

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.

1.6 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][6].

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.

1.7 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 jakéhokoliv 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!"}
```

1.8 Vzdálená komunikace mezi klientem a síťovým zařízením

1.8.1 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 využitě při přihlašování protokolem TELNET. Dále může útočník přihlášenému klientovi ukrást spojení a odeslat na server libovolný příkaz. Anglicky se tento útok nazývá „*session hijacking*“ [14].

1.8.2 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 [14].

1.9 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.

1.9.1 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 [10].

1.9.2 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[9], [10], [11].

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 pro MySQL od verze 5.0.3
- **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 značená 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 mohou být přiřazeny žádnému, jednomu, nebo více záznamům na druhé straně relace. V praxi se vazba relace typu více k více nahrazuje vložením další nové entity doprostřed relace, čímž vznikají dvě relace s vazbou typu jedna k více.
- Rekursivní relace – jsou relace, které vnikají mezi záznamy v jedné entitě. Mohou mít také vazbu jedna k jedné, jedna k více, nebo více k více [9], [10].

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 můžou 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 [9], [10], [11].

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 [12].

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 [12].

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 jejích 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 [12].

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*).

1.9.3 Programovací jazyk PHP

PHP patří mezi nejpoužívanější 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 *enginu PHP*⁹, ten je bude interpretovat. Kód PHP se od kódu odděluje pomocí určitých

⁹ Engine PHP je vykonavatel příkazů skriptovacího jazyka PHP.

značek proto, aby mohl *parser*¹⁰ okamžitě určit, které oblasti na stránce představují kód PHP. Prohlížeči webových stránek jsou po zpracování odesílány soubory, které neobsahují žádný zdrojový kód PHP [9].

¹⁰ 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 síťových zařízení Cisco

V této bakalářské práci jsem vytvořil komplexní systém 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 na eagle-serveru. Celý systém jsem pojmenoval *AutoConf*. Název *AutoConf* je zkratkou anglického 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 umístěných na eagle-serveru. Díky nim může uživatel sledovat stav spojení eagle-serveru 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 EXPECT skriptů, rovněž uložených na eagle-serveru. Po jejím úspěšném provedení je dané síťové zařízení nakonfigurováno pro měření laboratorní úlohy. Stačí jej už jen odpojit od sítě eagle-serveru a zapojit podle schématu zapojení dané laboratorní úlohy.

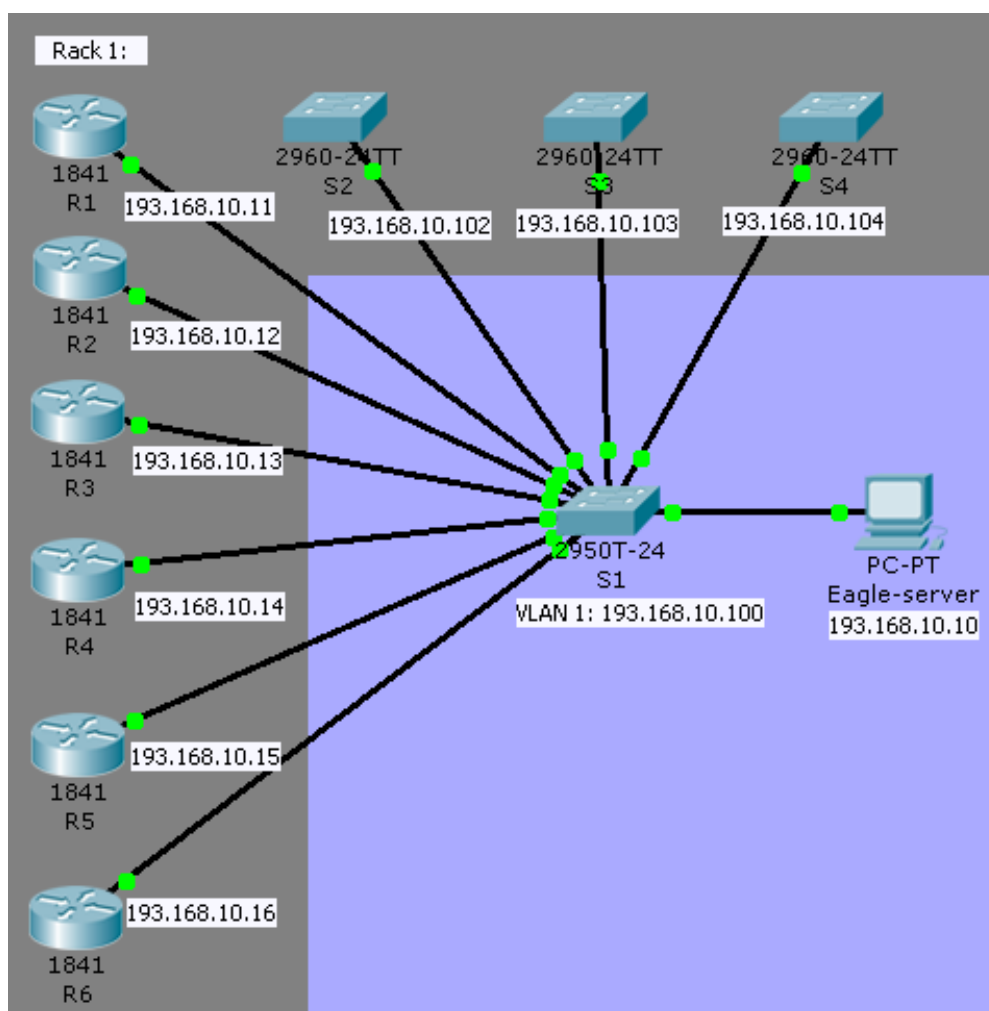
Všechny informace o připojených síťových zařízeních, laboratorních úlohách, případně vykonaných či nevykonaných konfiguracích, jsou přitom spravovány pomocí PHP funkcí a databáze MySQL. Konfigurační soubory jednotlivých laboratorních úloh se nacházejí přímo na eagle-serveru.

Aby bylo možné ovládat síťová zařízení pomocí systému *Autoconf*, musí být tato zařízení fyzicky připojena do sítě eagle-serveru. Obr. 3 ukazuje zjednodušené schéma zapojení sítě eagle-serveru, 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*¹¹ s tím rozdílem, že propojovací přepínač v každém dalším rozvaděči je propojen s eagle-serverem pomocí spojení s *hlavním přepínačem eagle-serveru*¹².

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 jednoduchým souborem uživatelů a uživatelských práv.



Obr. 3: Schéma zapojení zařízení pro automatickou konfiguraci

¹¹ Tvar fyzického propojení síťových zařízení.

¹² Přepínač, na který je přímo připojen eagle-server. Na **Chyba! Nenalezen zdroj odkazů.** značen jako S1.

2.2 Inicializace síťového zařízení v systému *AutoConf*

Síťová zařízení v učebně PA-128 jsou standardně nastavena tak, aby po svém zapnutí nahrála a spustila prázdný konfigurační soubor. To znamená, že nemají nastavené žádné síťové rozhraní ani přístupová hesla. Jejich síťové protokoly mají pouze základní nastavení. Aby bylo možné konfigurovat síťové zařízení z eagle-serveru, musí na něm být nastaveno alespoň jedno síťové rozhraní pro komunikaci s eagle-serverem. Mezi tímto rozhraním a eagle-serverem musí existovat fyzické spojení. Dále musí být povolená 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 eagle-serveru, heslo pro autorizaci spojení pomocí protokolu TELNET a heslo pro přechod do privilegovaného režimu jsou uloženy v databázi systému *AutoConf*.

Systém *AutoConf* nabízí možnost plné automatizace nastavení síťových zařízení. To znamená, že pokud se zařízením chceme komunikovat prostřednictvím webových stránek na eagle-serveru, tak se dané zařízení po svém zapnutí samo nastaví na komunikaci s eagle-serverem. Pokud jej ale pomocí eagle-serveru konfigurovat nechceme, nastaví se na něm prázdná konfigurace. Za tímto účelem jsem vytvořil EEM applet *restart* a TCL skript *autoconf.tcl*. EEM applet je indikátorem zapnutí síťového zařízení a TCL skript zajišťuje konfiguraci síťového zařízení na základě fyzického spojení s eagle-serverem.

V současné době je v učebně PA-128 instalace EEM appletu spolu s TCL skriptem možná pouze na směrovače, protože operační systém IOS přepínačů série *Cisco Switch Catalyst 2960* nepodporuje konfiguraci EEM ani TCL.

2.2.1 Detekce zapnutí směrovače pomocí EEM appletu

EEM applet má za úlohu spustit TCL skript po každém restartu směrovače. Detektorem události appletu je protokol SYSLOG, který monitoruje systémové zprávy směrovače. Po každém spuštění směrovače systém vypíše na výstup zprávu:

%SYS-5-RESTART: System restarted –

Na tuto zprávu zareaguje applet. Příkazem *enable* přejde do privilegovaného režimu směrovače a tam spustí TCL skript s názvem *autoconf.tcl*. Po úspěšném proběhnutí TCL skriptu vypíše na výstup zprávu „*KONFIGURATION DONE*” (Konfigurace proběhla).

Nastavení EEM appletu na síťovém zařízení

V systému *AutoConf* se EEM applet nahrává do paměti směrovačů v podobě startovací konfigurace obsahující pouze daný EEM applet s názvem *restart*. O nahrávání startovací konfigurace obsahující EEM applet, se stará EXPECT skript *configurator.exp*. Nahrávání startovací konfigurace je možné vykonat pomocí webových stránek eagle-serveru – kliknutím na akční tlačítko *EEM* v tabulce *SEZNAM ZAŘÍZENÍ* v modulu *zarizeni.php*.

Startovací konfigurace s EEM appletem

Startovací konfigurace pro nastavení automatické detekce zapnutí síťového zařízení pomocí EEM appletu je textový soubor s názvem *startup_router*, uložený na eagle-serveru v adresáři *soubory/nastaveni_zar*. Během procesu nahrávání EEM appletu na síťové zařízení se tento soubor kopíruje do paměti NVRAM síťového zařízení. Při startu síťového zařízení se pak automaticky nahraje do běžící konfigurace zařízení.

Soubor startovací konfigurace obsahuje příkazy pro základní nastavení síťového zařízení a sled příkazů pro nastavení EEM appletu. Při jeho nastavování je nutné definovat detektor spuštění EEM appletu a následně příkazy, které se mají po detekci vykonat.

```
scripting tcl encdir flash:
# Specifikace cílové cesty externích kódovacích souborů TCL do
# paměti flash.
event manager applet restart
# Vytvoření appletu s názvem restart.
event syslog occurs 1 pattern "%SYS-5-RESTART: System restarted"
# Nastavení detektoru spuštění appletu.
action 1.0 cli command "enable"
# Příkaz na cli: Přejít do privilegovaného režimu.
action 2.0 cli command "tclsh flash:autoconf.tcl"
# Spuštění TCL skriptu
action 3.0 syslog msg "CONFIGURATION DONE"
# Zpráva na výstup: Konfigurace provedena.
```

2.2.2 TCL skript pro konfiguraci směrovače

TCL skript s názvem *autoconf.tcl* je uložen v paměti FLASH v každém síťovém zařízení, ve kterém chceme vykonávat automatické nastavení spojení s eagle-serverem. Za pomoci EEM appletu *restart* je spuštěn po každém restartu zařízení.

Jeho úlohou je zjistit, zda si uživatel přeje nahrát na síťové zařízení konfigurační soubor z eagle-serveru. Skript zajistí, aby síťové zařízení určitou dobu po svém zapnutí otestovalo spojení jeho ethernetového rozhraní *f0/1* s přepínačem, na kterém je připojený

eagle-server. Pokud test spojení *ping*¹³ proběhne úspěšně, TCL skript nastaví vše potřebné pro komunikaci síťového zařízení s eagle-serverem. Za tímto účelem je potřeba v přepínači nastavit rozhraní *VLAN 1*¹⁴ s IP adresou, kterou budou síťová zařízení testovat.

Pokud by uživatel nechtěl konfigurovat síťové zařízení z eagle-serveru, stačí, aby jej před jeho restartem odpojil od přepínače eagle-serveru. TCL skript by na základě neúspěšnosti operace *ping* nastavil na směrovači prázdnou běžící konfiguraci.

Síťová zařízení testují spojení s přepínačem, na který je připojený eagle-server, a ne přímo s eagle-serverem. Počítač, na kterém pracuje eagle-server, nemusí být v čase inicializace síťových zařízení zapnutý, protože přepínač eagle-serveru se vždy zapíná současně se všemi síťovými zařízeními.

Konfigurace TCL skriptu *autoconf.tcl*

Systém *AutoConf* poskytuje možnost automatizované konfigurace TCL skriptu na síťová zařízení pomocí webové aplikace. Použitím akčního tlačítka *TCL* v tabulce *SEZNAM ZAŘÍZENÍ* v modulu *zarizeni.php* se spustí PHP skript *flash.php*. Ten vytvoří zdrojový kód TCL skriptu a uloží jej do adresáře *soubory/flash* na eagle-server. TCL skript je na eagle-serveru uložen pod názvem bez diakritiky ve formátu *autoconf_jmeno_zarizeni.tcl*. Pomocí EXPECT skriptu *flash.exp* je takto vytvořený TCL skript odeslán na dané síťové zařízení. Kvůli funkcionalitě appletu *restart* se TCL skript ukládá pod názvem *autoconf.tcl* do paměti FLASH síťového zařízení.

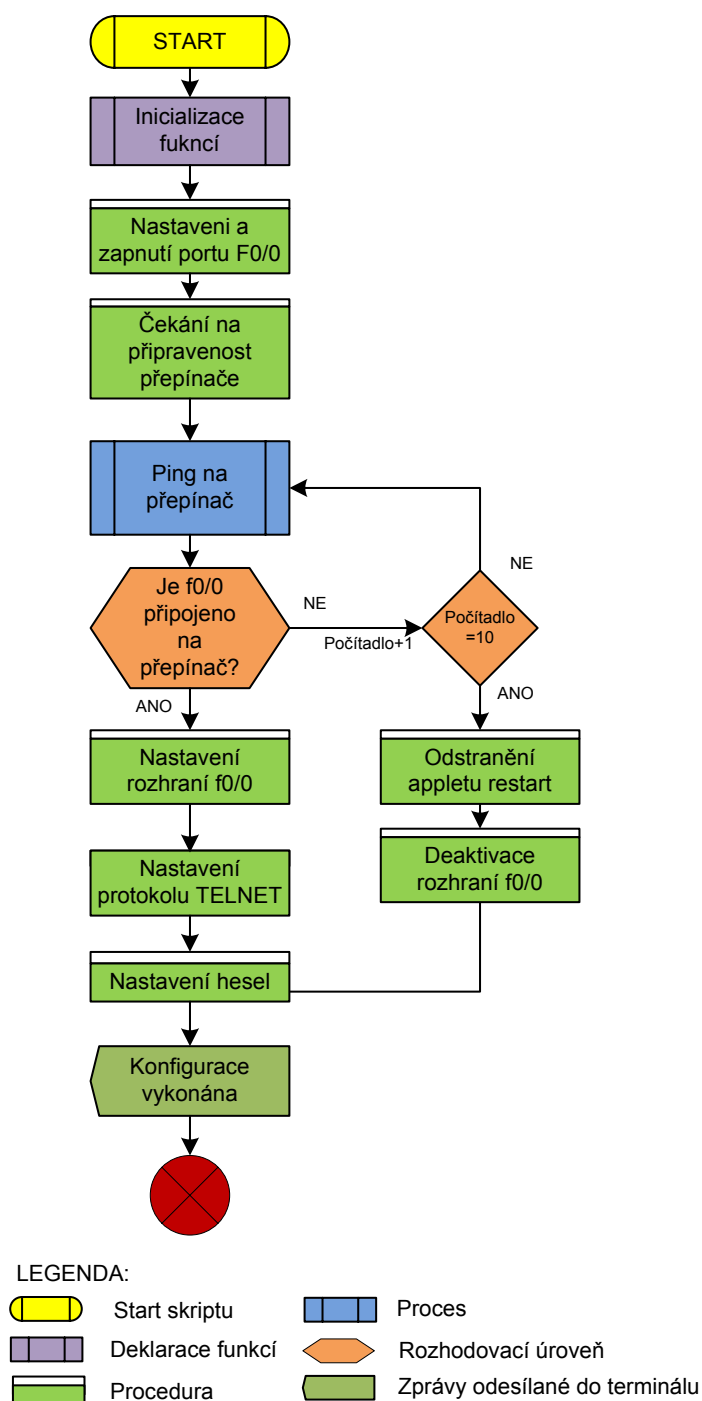
TCL skript *autoconf.tcl* je nutné vytvořit pro každé síťové zařízení zvlášť. Obsahuje totiž informace o tom, jak má být dané síťové zařízení nastaveno tak, aby bylo schopné připojit se k eagle-serveru, a systém *AutoConf* jej na základě údajů uložených ve své databázi identifikoval správně.

Popis běhu TCL skriptu *autoconf.tcl*:

Průběh inicializace směrovače, vykonané pomocí TCL skriptu *autoconf.tcl* a odstartované pomocí EEM appletu *restart*, popisuje vývojový diagram na Obr. 4.

¹³ Příkaz, který umožňuje prověřit funkčnost spojení.

¹⁴ Virtuální rozhraní, do kterého mají přístup všechny porty přepínačů ve standardním nastavení.



Obr. 4: Práce TCL skriptu *autoconf.tcl*

- 1) Po spuštění *autoconf.tcl*, skript nejdříve inicializuje všechny své funkce a nastaví potřebné proměnné. Pak je už jen postupně uvádí do chodu.
- 2) Spustí se samotný program.
- 3) Nastaví se IP adresa na rozhraní *fast-ethernet f0/1* a rozhraní se zapne.
- 4) Spustí se časovač, který čeká 90 sekund. Je to proto, že start přepínače eagle-serveru trvá déle jako start směrovače, a proto zařízení čeká, až bude přepínač připraven.

- 5) Spustí se test spojení *ping*, který se provádí dokud není úspěšný, maximálně však desetkrát.
- 6) Pokud test proběhne úspěšně, nakonfiguruje se rozhraní *f0/1*, služba TELNET, heslo pro přístup na TELNET a heslo pro přechod do privilegovaného režimu.
- 7) Pokud se test nepodaří, odstraní se applet *restart* a rozhraní *f0/1* se deaktivuje.
- 8) TCL skript *autoconf.tcl* končí.
- 9) Applet *restart* vypíše na výstup zprávu: „Konfigurace vykonána.“ a skončí.
- 10) Konec procesu inicializace síťového zařízení.

Důležité parametry TCL skriptu *autoconf.tcl*:

- Parametry pro nastavení síťového zařízení – IP adresa, na kterou má být nastaveno ethernetové rozhraní síťového zařízení; dále heslo pro autorizaci TELNET spojení a heslo pro přechod do privilegovaného režimu. Tyto parametry jsou definované záznamem uloženým v entitě *ZARIZENI* databáze MySQL.
- IP adresa síťového zařízení, se kterým se bude testovat spojení během inicializace síťového zařízení – ve standardním nastavení systému *AutoConf* je to IP adresa přepínače *S1*, který je připojen k eagle-serveru (viz Obr. 3). Je definovaná v PHP funkci *nastaveni()* v PHP skriptu *ssh_spojeni.php*.
- Obsah proměnné *wait*¹⁵ – tato proměnná obsahuje časový údaj v nanosekundách o tom, jak dlouho má síťové zařízení čekat, než začne testovat spojení s přepínačem eagle-serveru.
- Obsah proměnné *counter*¹⁶ – v proměnné *counter* je definováno, kolikrát se má TCL skript pokusit provést test spojení s přepínačem eagle-serveru *ping* než bude spojení úspěšné. Parametry *watt* a *counter* je možné nastavovat pouze ručně editací PHP funkce *autoconf()* v PHP skriptu *flash.sh*.

¹⁵Pro optimalizaci nastavení systému *AutoConf* v učebně PA-128 byla hodnota proměnné *watt* nastavena na 90000 ns.

¹⁶ Pro optimalizaci nastavení systému *AutoConf* v učebně PA-128 byla hodnota proměnné *counter* nastavena na 10.

2.2.3 Inicializace přepínače v systému *Autoconf*

Původně měla být inicializace přepínačů v systému *AutoConf* prakticky stejná jako inicializace směrovačů. To znamená, že měl být použit EEM applet pro detekci zapnutí přepínače a spuštění TCL skriptu. TCL skript pak měl testovat spojení s přepínačem eagle-serveru nebo přímo s eagle serverem. Na základě výsledku testu spojení měl být přepínač následně nastaven pro komunikaci s eagle-serverem nebo s prázdnou konfigurací.

V učebně PA-128 jsou bohužel přepínače série *Cisco Switch Catalyst 2960* se starší verzí operačního systému IOS, která nepodporuje konfiguraci EEM ani TCL. Proto jsem se rozhodl systém *Autoconf* překonfigurovat tak, aby umožňoval alespoň základní natavení inicializace přepínače do systému *AutoConf*. To znamená, že ve startovací konfiguraci přepínačů v učebně PA-128 je uložena konfigurace, která pokaždé nastaví přepínač přímo na konfiguraci z eagle-serveru. Neexistuje přitom žádný rozhodovací mechanismus.

Konfigurace přepínače pro spolupráci se systémem *AutoConf*

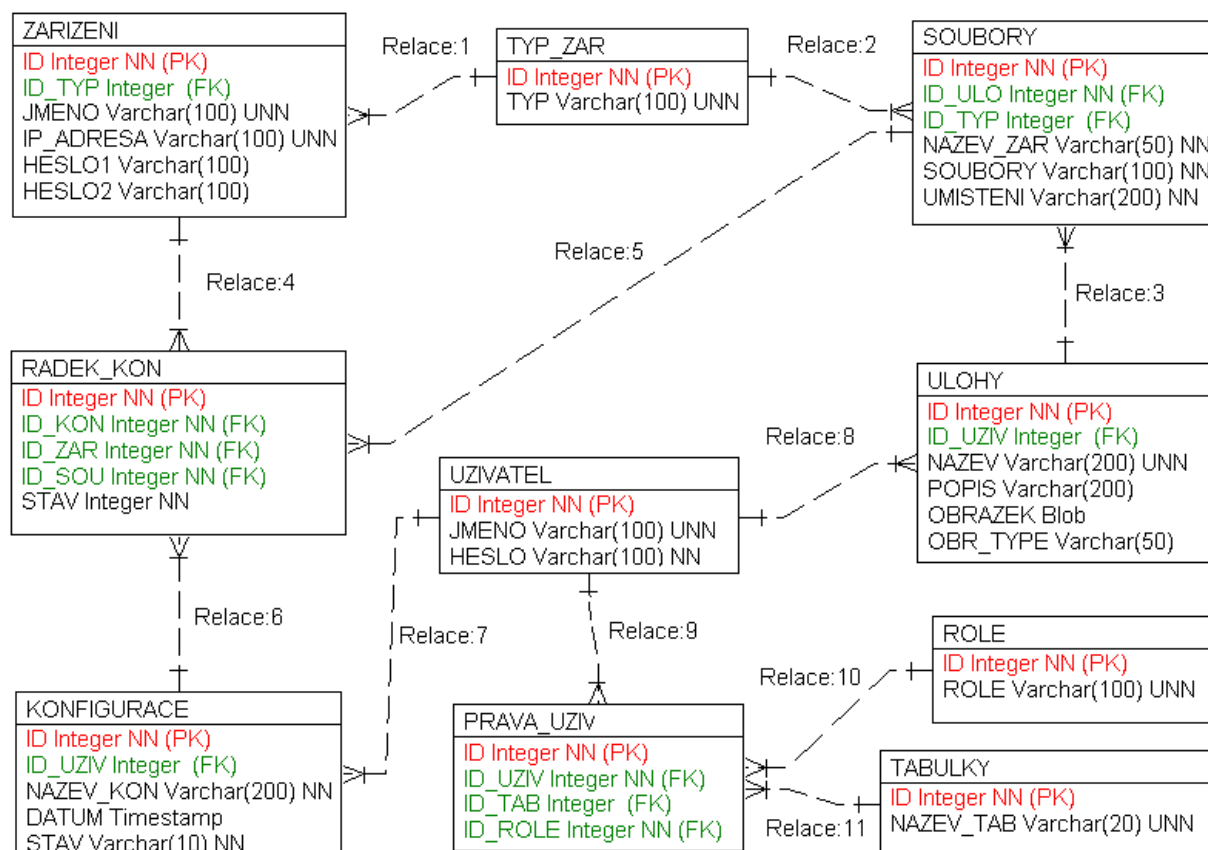
Systém *AutoConf* umožňuje automatické nastavení startovací konfigurace přepínače. Po kliknutí na akční tlačítko *STU* v tabulce *SEZNAM ZAŘÍZENÍ* PHP modulu *zarizeni.php*, systém vygeneruje zdrojový kód startovací konfigurace pro daný přepínač. Následně je tato konfigurace zkopírována do daného přepínače.

2.3 *Autoconf* – databáze systému pro automatickou konfiguraci

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

- Záznamy o zařízeních, které se budou konfigurovat pomocí eagle-serveru.
- 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: Entitně – relační diagram databáze *autoconf*

Entita *ZARIZENI* představuje seznam všech směrovačů a přepínačů, 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 přepínači konfigurační soubor

určený pro směrovač a opačně. Směrovači nejde přiřadit konfigurační soubor určený pro přepínač.

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 pouze 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 *Učitel* a *Student*. Uživatel, který je přihlášen jako *Učitel*, 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. Login: Učitel Heslo: Teacher
2. Přidat	Ano	Ano	–	–	
3. Upravit	Ano	Ano	Ano	–	
4. Nova kon	–	–	Ano	Ano	
5. Vykonat	–	–	Ano	–	
6. Prohlednout	Ano	Ano	Ano	Ano	
7.Rychla kon	–	–	–	Ano	
8. Nastav zar	Ano	–	–	–	
1. Mazat	Ne	Ano	Ne	–	ID: 2. Login: Student Heslo: Student
2. Přidat	Ne	Ano	–	–	
3. Upravit	Ne	Ano	Ne	–	
4. Nova kon	–	–	Ano	Ano	
5. Vykonat	–	–	Ano	–	
6. Prohlednout	Ano	Ano	Ano	Ano	
7.Rychla kon	–	–	–	Ne	
8. Nastav zar	Ano	–	–	–	

Tab. 2: Popis atributů v entitách databáze *Autoconf*

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	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.
SOUBORY	ID ID_ULO ID_TYP NAZEV_ZAR SOUBORY UMISTENI	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. Umístění konfiguračního souboru na eagle-serveru.
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 *AutoConf*

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 eagle-serveru pomocí SSH, *bash.php* – spouští skript *tester.exp* pro test spojení a kontrolu nastavení mezi eagle-serverem a síťovým zařízením, *inicializace.php*, *bashrychle.php*, *startup.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.
- 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.
- Knihovna pro dynamizaci webových stránek na straně klienta: *prototype.js* – Tato knihovna 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 [16].

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.
$mysqlldb = „Autoconf“;
// Vytvoření nového konstrukturu spojení s databází MySQL serveru.
$mysqli = @new mysqli($mysqlhost, $mysqluzivatel, $mysqlheslo,
$mysqlldb);
// 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 konstrukturu 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 eagle-serveru byla vytvořena funkce *ssh2_pripoj()*, která vrací konstruktor spojení PHP s SSH klientem eagle-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_connec`, 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.

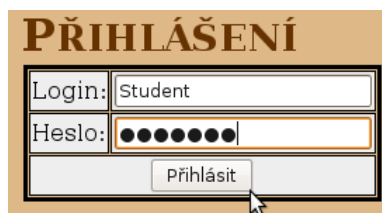
```
function ssh2_pripoj() {
// Cyklus umožňující dva pokusy o navázání SSH spojení.
for($pocitadlo=0; $pocitadlo<"2"; $pocitadlo++){
    // Kontrola existence spojení PHP s SSH.
    if(!($connection = @ssh2_connect("localhost", 22)) and
    $pocitadlo == "1"){
        // Výpis a ukončení funkce pokud 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 konstrukturu spojení k SSH.
return $connection;
}
```

2.4.3 Autorizace uživatelů do systému *AutoConf*

Autorizace 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 do systému *AutoConf*

Skript *42řihlásit.php* je vstupním přihlašovacím formulářem pro přístup uživatele do systému *AutoConf*. 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*.



Obr. 6: Přihlašovací okno systému *AutoConf*

Při neplatném přihlášení, odhlášení uživatele, nebo vypršení doby nečinnosti uživatele v systému, je uživatel automaticky přesměrován do modulu *43řihlásit.php*.

Kontrola autorizace uživatele v modulech systému *AutoConf*

Skript *kontrola.php* zabezpečuje kontrolu uživatelského přístupu do 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 *43řihlásit.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 49). 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*.

SEZNAM ZAŘÍZENÍ

Typ	Název	IP adresa	Telnet heslo	# heslo	Stav	Možnosti
Router	Rack1-R1	193.168.10.11	cisco	cisco	Připojeno!	Upravit Smazat EEM TCL
Router	Rack1-R2	193.168.10.12	cisco	cisco	Odpojeno!	Upravit Smazat EEM TCL
Switch	Rack1-S2	193.168.10.22	cisco	cisco	Testuji. . .	Upravit Smazat STU

Obr. 7: Ukázka seznamu zařízení určených pro automatickou konfiguraci

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 tlačítka *EEM*, *TCL* a *STU*. Akční tlačítko *EEM* umožňuje uživateli uložit do směrovače startovací konfiguraci s EEM appletem *restart*, potřebným pro automatické nastavení směrovače. Akční tlačítko *TCL* kopíruje TCL skript *autoconf.tcl* na směrovače do paměti FLASH. Funkce appletu *restart* a skriptu *autoconf.tcl* byly popsány v kapitole 2.2. *STU* nahraje do přepínače startovací konfiguraci, která přepínač hned po jeho zapnutí nastaví pro komunikaci se systémem *Autoconf*. Je to kvůli tomu, že přepínače nacházející se v laboratoři PA-128, nepodporují konfiguraci appletů ani TCL skriptů.

Tab. 3: Seznam všech možných stavů síťového zařízení v systému *AutoConf*

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 v systému *Autoconf*

Před návrhem nové laboratorní úlohy musíme mít nejdříve na klientském počítači uloženy konfigurační soubory pro danou 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.

Editace 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. Do databáze se ukládá pouze název konfiguračního souboru. Samotné konfigurační soubory jsou ukládány na eagle-server do adresáře *SOUBORY* pomocí mechanismu HTTP POST skriptovacího jazyka PHP.

Pokud ale nastane situace, kdy uživatel chce úlohu nebo konfigurační soubor změnit či vymazat, PHP funkce volají příkazy BASH pro přejmenování nebo mazání souborů, popsané v kapitole 2.5.2.

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.

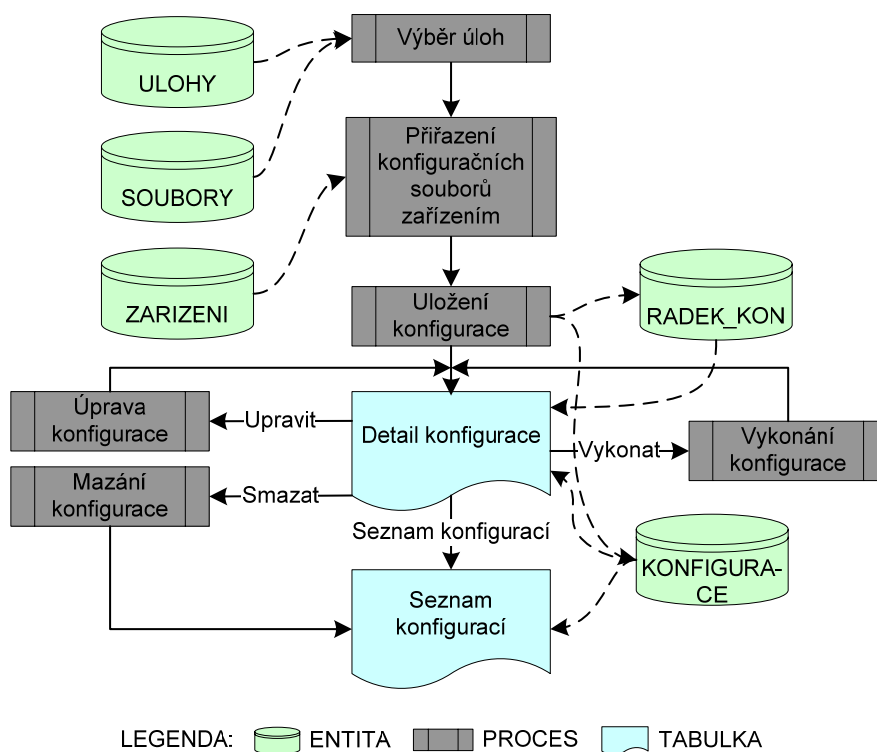
2.4.6 Vytváření a vykonávání konfigurací v systému *AutoConf*

Úlohou skriptu *konfigurace.php* je vytváření a vykonávání konfigurací na síťových zařízeních. Obr. 9 demonstruje postup, jakým se ve skriptu *konfigurace.php* vyrábí nové konfigurace, které je možné následně vykonat, případně editovat. Vytváření nových konfigurací v systému *AutoConf* je proces, při kterém se spojují do dvojice identifikátor 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*.

Postup pro vytvoření nové konfigurace:

- Výběr úloh s konfiguračními soubory ze seznamu všech úloh databáze *Autoconf*.
- Pojmenování nové konfigurace a přiřazení konfiguračních souborů zařízením.
- Uložení nové konfigurace do databáze.
- Vykonání části nebo celé konfigurace.
- Sledování úspěšnosti konfigurace jednotlivých síťových zařízení a také celé konfigurace.

Při zobrazení detailu vytvořené konfigurace (viz Obr. 10) 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. Úspěšně nakonfigurovat lze ovšem pouze zařízení, která jsou připojena.



Obr. 9: Postup vytváření konfigurace v systému *AutoConf*

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).

Hlavní nabídka: Seznam zařízení Seznam úloh Seznam konfigurací Nabídka konfigurace: Seznam konfigurací Konfigurovat vše Konfigurovat zašknuté Upravit Smazat						
KONFIGURACE: KONFIGURACE Č. 1						
	Typ	Název	IP adresa	Úloha	Konfigurační soubor	Stav
☐	Router	Rack1-R1	193.168.10.11	Lab 3.4.2 Managing a Web Server	R1-ISP	Odpojeno!
☒	Router	Rack1-R2	193.168.10.12	Lab 3.4.2 Managing a Web Server	R2-Central	Hotovo!
☒	Switch	Rack1-S2	193.168.10.22	Lab 3.4.2 Managing a Web Server	S1-Central	Konfiguruji. . .

Obr. 10: Ukázka konfigurace vytvořené v systému *AutoConf*

2.4.7 Seznam konfigurací uložených v databázi systému *AutoConf*

Správu nad všemi konfiguracemi v systému *AutoConf* má na starost modul *seznam_kon.php*. Uživateli poskytuje informace o existujících konfiguracích uložených v databázi a zpřístupňuje další moduly pro vytváření a vykonávání konfigurací.

Obr. 11 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	2009-05-25 19:28:26	Učitel	100 %	Prohlédnout
Konfigurace č. 2	2009-05-25 19:30:21	Učitel	40 %	Prohlédnout

Obr. 11: Ukázka seznamu existujících konfigurací

2.4.8 Modul pro rychlou konfiguraci síťového zařízení

Rychlá konfigurace (viz. Obr. 12) 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 *Učitel*.

RYCHLÁ KONFIGURACE					
Typ	Název	IP adresa	Stav	Konfigurační soubor	Moznosti
Router	Rack1-R1	193.168.10.11	Hotovo!	R1-ISP	Konfiguruj
Router	Rack1-R2	193.168.10.12	Konfiguruji...	R2-Central	Konfiguruj
Router	Rack1-R3	193.168.10.13	Odpojeno!	--Vyberte konfigurační soubor!--	Konfiguruj
Switch	Rack1-S2	193.168.10.22	Připojeno!	--Vyberte konfigurační soubor!--	Konfiguruj
Switch	Rack1-S3	193.168.10.23	Odpojeno!	--Vyberte konfigurační soubor!--	Konfiguruj
Switch	Rack1-S4	193.168.10.24	Odpojeno!	Lab 3.4.2 Managing a Web Server S1-Central Prázdná konfigurace Switch prázdná	Konfiguruj

Obr. 12: Rychlá konfigurace síťových zařízení

2.5 Skripty pro terminál eagle-serveru

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

- *tester.exp* – testuje spojení eagle-serveru a síťového zařízení.
- *flash.exp* – nahrává TCL skript do paměti FLASH síťového zařízení.
- *configurator.exp* – do síťových zařízení kopíruje startovací a běžící konfigurace.
- *startup_sw.exp* – nahrává do přepínačů startovací konfiguraci s nastavením pro připojení do systému *AutoConf* po startu přepínače.

2.5.1 Komunikace eagle-serveru se síťovým zařízením pomocí EXPECT skriptu

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

Postup konfigurace síťového zařízení pomocí EXPECT skriptu

Obr. 13 je vývojový diagram EXPECT skriptů *configurator.exp*, *flash.exp* a *tester.exp*. Všechny tyto skripty mají skoro stejný zdrojový kód. Musely být rozděleny do více souborů, protože skriptovací jazyk EXPECT nemá funkci pro větvení programu pomocí podmínky *if*.

Skripty používají stejný způsob připojení k síťovému zařízení, avšak *tester.exp* se hned po navázání spojení odpojí a skončí, *configurator.exp* pokračuje v konfigurování běžící nebo startovací 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.
- Začátek procesu odesláním příkazu *spawn telnet \$IP*. V proměnné *\$IP* je uložena IP adresa síťového zařízení.
- Čekání na odpověď se žádostí hesla pro autorizaci TELNET spojení „*password*”.

- Pokud proces odpoví „*Unable to connect*“, případně nepřijde odpověď do uplynutí času v proměnné *timeout*, skript skončí s hlášením: „*Odpojeno!*“
- Pokud proces odpoví „*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 z procesu nepřijde 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 v proměnné *\$PRIVILEG* pro přechod do privilegovaného režimu.
- Čekání na odpověď obsahující příznak přihlášení do uživatelského režimu „#“
- Pokud z procesu nepřijde odpověď do uplynutí času v proměnné *timeout* nebo je odpovědí zpráva „*Bad password*“, skript skončí s hlášením „*Chybné heslo #!*“
- Pokud proces vrátí text 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*“ nebo „*startup-config*“ pro nahrávání konfiguračních souborů z eagle-serveru na síťové zařízení ve skriptu *configurator.exp*, nebo „*flash*“ pro nahrání TCL skriptu do paměti FLASH v síťovém zařízení.
- Dále jsou odeslány parametry pro kopírování souboru z eagle-serveru. *\$ZDROJ* obsahuje IP adresu eagle-serveru jako zdroje souboru pro kopírování. Parametr v proměnných *\$UMISTENI**\$SOUBOR* označuje soubor a jeho úložiště na eagle-serveru.
- 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 z eagle-serveru na síťové zařízení.
- Pokud procesu 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*.

2.5.2 Příkazy BASH, volané pomocí PHP skriptu systému *AutoConf*

Při práci se systémem *Autoconf* se pro změnu názvu nebo mazání konfiguračních souborů uložených na eagle-serveru používají BASH příkazy. Protože při ukládání konfiguračních souborů nejsou nastavena uživatelská práva pro editaci souborů uživatelem nastaveným pro spojení SSH, musí být BASH příkazy pro práci s nimi vykonávány uživatelem *superuser*.

Přejmenování souboru pomocí BASH příkazu

```
sudo /bin/mv "původní název souboru" "nový název souboru"
```

Mazání souboru nebo adresáře pomocí BASH příkazu

```
sudo /bin/rm -rv "$UMISTENI"
```

V případě nezdařilého pokusu o provedení operace mazání, případně přejmenování souboru, systém *AutoConf* uloží nezdařilý příkaz do souboru *error.log* v adresáři */soubory*. Při údržbě systému je pak možné tyto příkazy manuálně vykonat.

Abychom nemuseli při vykonávání BASH příkazu zadávat heslo pro autorizaci super-uživatele, musíme editovat soubor */etc/sudoers*, který definuje práva uživatelů v operačním systému Linux. Do souboru */etc/sudoers* přidáme řádek

```
Jméno_uživatele ALL=(ALL) NOPASSWD: /bin/rm, /bin/mv, umístění_adresáře_SOUBORY.
```

2.6 Instalace systému *AutoConf*

2.6.1 Potřebné softwarové nástroje

Pro instalaci systému *Autoconf* na eagle-server 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 eagle-serveru pomocí zabezpečeného spojení SSH.
- Balíček EXPECT – pomocí něj je možné spouštět EXPECT skripty v příkazovém řádku eagle-serveru.
- TFTP server – umožňuje kopírování konfiguračních souborů z eagle-serveru na síťové zařízení. Je potřeba nastavit TFTP server pro možnost zadávání úplné adresy konfiguračního souboru, určeného pro kopírování.

2.6.2 Nastavení systému *AutoConf*

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 *System_AutoConf*, který obsahuje všechny skripty systému *AutoConf*, je potřeba zkopírovat do paměti eagle-serveru.

Pro umožnění komunikace PHP funkcí s databází MySQL a s terminálem eagle-serveru 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 přepínače eagle-serveru. TCL skript na ní bude provádět  
    $konstanta["test"] = "193.168.10.10";           // test spojení.  
    // IP adresa eagle-serveru. Využívá se při kopírování konfiguračních  
    // souborů z eagle-serveru na síťové zařízení.  
    $konstanta["zdroj"] = "193.168.10.100";  
    // Úplná adresa adresáře, ve kterém se nacházejí konfigurační  
    // soubory pro síťová zařízení.  
    $konstanta["adresar_souboru"]="umístění adresáře soubory/";  
    // Úplná adresa adresáře, ve kterém se nacházejí skripty pro  
    // komunikaci eagle-serveru se síťovými zařízeními.  
    $konstanta["adresar_bash"] = "umístění adresáře bash/";  
    return $konstanta;  
}
```

Základní instalace systému *AutoConf* obsahuje i záznamy o přepínačích a směrovačích, na kterých byl systém *AutoConf* testován, dále pak úlohu obsahující prázdné konfigurace pro přepínače a směrovače a laboratorní úlohu z kurzu CCNA 1.

2.6.3 Softwarové nástroje, použité při testování

- Ubuntu 8.10 – distribuce operačního systému Linux, použita na eagle-serveru.
- 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 bakalářské 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, které se nacházejí v učebně Cisco akademie.

Systém je řízen pomocí webových stránek, umístěných na eagle-serveru, specifickém počítači v laboratoři. Pomocí webových stránek může uživatel vkládat do systému nové laboratorní úlohy a pak jejich konfigurační soubory přidělovat síťovým zařízením, připojeným do systému *AutoConf*. Následně je pak možná jejich konfigurace. Informace o síťových zařízeních, se kterými systém *AutoConf* pracuje, 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 eagle-server. Mechanismus pak toto spojení detekuje a nastaví vše potřebné pro komunikaci daného směrovače s eagle-serverem. V opačném případě se na směrovači nastaví prázdná konfigurace. Mechanismus rozhodování je však možné aplikovat pouze na směrovače připojené k eagle-serveru.

Přepínače, které se v současné době nacházejí v laboratoři Cisco akademie, nepodporují nastavení EEM ani TCL, potřebné pro aplikaci mechanismu rozhodování. Systém *AutoConf* na nich ovšem umožňuje vytvořit startovací konfiguraci pro spojení přepínače s eagle serverem. Startovací konfigurace způsobí, že se přepínač nastaví vždy tak, aby mohl pracovat se systémem *AutoConf*. Stačí jej pouze fyzicky propojit s přepínačem, na který je připojen eagle-server. Nastavení startovací konfigurace na připojeném přepínači se provádí pomocí webových stránek systému *AutoConf*.

V rámci této bakalářské práce byl také vytvořen jednoduchý 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.

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] *Ping* [online]. Wikipedia, 2008 [cit. 2008-12-15]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Ping>>.
- [6] MILAR, Bohdan. *Seriál o BASHi* [online]. 2005 [cit. 2008-12-15]. Dostupný z WWW: <<http://www.linuxexpres.cz/praxe/serial-o-bashi>>.
- [7] *Seriál TCL základy* [online]. 2004 [cit. 2008-12-15]. Dostupný z WWW: <<http://www.root.cz/serialy/tcl-zaklady/>>.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] KOFLER, Michael . *Mistrovství v MySQL 5 : Kompletní průvodce webového vývojáře*. 1. vyd. Brno : Computer press, 2007. 808 s. ISBN 978-80-251-1502-2.
- [12] 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.

- [13] MACH, Jakub. *PHP pro úplné začátečníky* . 2. rozš. vyd. Brno : Computer Press, 2006. 167 s. ISBN 80-251-1248-9.
- [14] VELTE, Toby J., VELTE, Anthony T. *Síťové tefnologie CISCO : Velký průvodce*.
- [15] David Krásenský. Brno : Computer Press, 2003. ISBN 80-7226-857-0. s. 0-759. Cisco akademie, *UTP kabeláž v laboratoři* [on line]. c2008 [cit. dne 20.3.2009]. Dostupné z http://adela.utko.feec.vutbr.cz/cisco_akademie/index.php?option=com_content&task=view&id=17&Itemid=33.
- [16] STEPHENSON, Sam. *Prototype JavaScript framework* [online]. 2006-2007 [cit. 2009-03-20]. Stránky pro ztažení 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	– Random-Access Memory
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

Seznam příloh

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

A Stromová struktura adresáře *System_AutoConf*

ajax_lib/
 prototype.js
 readme.txt

autorizace/
 autorizace.php
 kontrola.php

bash/
 configurator.exp
 flash.exp
 startup_sw.exp
 tester.exp

db_skripty/
 db_funkce.php
 db_spojeni.php

phpkbash/
 bash.php
 bashrychle.php
 flash.php
 inicializace.php
 ssh_spojeni.php
 startup.php

soubory/
 flash/
 autoconf_rack1-r1
 lab_3.4.2_managing_a_web_server_2/
 r1-isp
 r2-central
 s1-central
 nastaveni_zar/
 startup_router
 prazdna_konfigurace_1/
 router_prazdna
 switch_prazdna
 error.log

vzhled/
 hlavicka.php
 nabidka_vlevo.php
 styly.css
 zadny.gif

konfigurace.php
php_funkce.php
prihlasit.php
rychla_kon.php
seznam_konf.php
ulohy.php
zarizeni.php
zobrazobr.php

B Obsah přiloženého CD

Bakalařska_prace.pdf	– bakalářská práce
Autoconf.sql	– zdrojový kód databáze MySQL
System_AutoConf	– adresář obsahující zdrojové kódy systému <i>AutoConf</i>