

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2019

Bc. Petr Jašíček



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

SOFTWAREVÝ KONTROLÉR WI-FI PŘÍSTUPOVÝCH BODŮ SE SYSTÉMEM OPENWRT

SOFTWARE CONTROLLER FOR OPENWRT BASED WI-FI ACCESS POINTS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Petr Jašíček

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Ilgner

BRNO 2019

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Petr Jašíček

ID: 152656

Ročník: 2

Akademický rok: 2018/19

NÁZEV TÉMATU:

Softwarový kontrolér Wi-Fi přístupových bodů se systémem OpenWrt

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a implementovat kontrolér pro centralizovanou správu bezdrátových přístupových bodů v sítích IEEE 802.11 se systémem OpenWrt. Realizovaný kontrolér umožní uživateli vytvořit ucelenou síť, jejíž základní správu a dohled je možné provádět z webového prostředí kontroléru. Vytvořený kontrolér umožní automatizované nasazení konfigurace přístupových bodů (IP, NTP server, parametry bezdrátové sítě a jejího zabezpečení, konfigurace VLAN a SSID). Kontrolér bude naopak z přístupových bodů získávat informace o jejich stavu (seznam asociovaných klientů, úroveň signálu, zatížení zařízení, uptime, verze softwaru). Kontrolér zajistí snadnou adaptaci nového zařízení. Navrhněte a implementujte zabezpečení komunikace mezi kontrolérem a spravovanými zařízeními.

Uživatelské prostředí aplikace kontroléru bude realizováno jako webové s využitím technologií dostupných na platformě OpenWrt (Python, Flask, Lua). Případné úpravy softwaru přístupových bodů musí být realizovány jako opkg balíček. Backend operace implementujte v jazyce Python, Bash, C.

DOPORUČENÁ LITERATURA:

[1] OpenWrt Documentation [online], 2018. Dostupné z: <https://openwrt.org/docs/start>

[2] PILGRIM, Mark. Ponořme se do Python(u) 3: Dive into Python 3. Praha: CZ.NIC, z.s.p.o. ISBN 978-8-904248-2-1.

Termín zadání: 1.2.2019

Termín odevzdání: 16.5.2019

Vedoucí práce: Ing. Petr Ilgner

Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Cílem práce bylo vytvoření webové aplikace určené pro centralizovanou správu a konfiguraci bezdrátových sítí na přístupových bodech běžících na distribuci OpenWrt. Čtenář je v úvodu seznámen s obecnou problematikou bezdrátových sítí, včetně seznámení s existujícími řešeními jejich centralizované správy. Následuje seznámení s distribucí OpenWrt a způsobu konfigurace pomocí modelu UCI. Dále jsou popsány hlavní technologie použité v této práci. Následuje popis návrhu architektury, uživatelského rozhraní a způsobu implementace výsledné webové aplikace. Výstupem práce je platformě nezávislá webová aplikace, která umožňuje jednoduchou správu a konfiguraci přístupových bodů se systémem OpenWrt.

KLÍČOVÁ SLOVA

OpenWrt, IEEE 802.11, JSON-RPC, UCI, konfigurace, VLAN, Flask, Python

ABSTRACT

The aim of this thesis was to create a web application used for centralized management and configuration of wireless networks on access points running on OpenWrt distribution. The reader is acquainted with common principles of wireless networks including description of existing solutions for their centralized management. After that the thesis introduces the reader to OpenWrt distribution and to a method of its configuration using UCI model. The reader is then introduced to the main technologies used in this thesis. Author further describes architecture, user interface and implementation of the web application. The created web application is platform agnostic and provides easy way to manage and configure access points with OpenWrt distribution.

KEYWORDS

OpenWrt, IEEE 802.11, JSON-RPC, UCI, configuration, VLAN, Flask, Python

JASÍČEK, Petr. *Softwarový kontrolér přístupových bodů Wi-Fi se systémem OpenWRT*. Brno, Rok, 59 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: prof. Ing. Petr Ilgner,

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Softwarový kontrolér přístupových bodů Wi-Fi se systémem OpenWRT“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Petru Ilgnerovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora

Obsah

Úvod	11
1 Lokální bezdrátové sítě	12
1.1 Bezdrátové sítě dle standardu IEEE 802.11	12
1.1.1 Sady služeb	13
1.2 Konfigurace bezdrátových přístupových bodů	14
1.2.1 UniFi SDN	15
1.2.2 Platforma OpenWISP	16
2 Linuxová distribuce OpenWrt	17
2.1 Kompilace	17
2.2 Instalace	18
2.3 První přihlášení	19
2.4 Práce s balíčky	19
2.5 LuCI	20
3 Konfigurační model OpenWrt	21
3.1 Unifikované konfigurační rozhraní	21
3.1.1 Syntaxe	22
3.1.2 Konfigurace nástrojem uci	23
3.1.3 Konfigurace přes rozhraní JSON-RPC	23
4 Použité technologie	26
4.1 Webový aplikační framework Flask	26
4.2 JavaScriptová knihovna jQuery	30
5 Návrh řešení	31
5.1 Role kontroléru	31
5.2 Formulace požadavků	32
5.2.1 Definování bezdrátových sítí	32
5.2.2 Zobrazení stavu zařízení	33
5.3 Návrh uživatelského rozhraní	35
5.3.1 Přihlašovací stránka	35
5.3.2 Hlavní stránka	35
5.3.3 Detail zařízení	37
5.3.4 Definice lokálních sítí	38
5.3.5 Definice bezdrátových sítí	39
5.3.6 Seznam asociovaných klientů	40

6 Implementace	42
6.1 Datový model	42
6.2 Přihlášení uživatele do kontroléru	42
6.3 Autentizace k zařízení OpenWrt	43
6.4 Skenování sítě	45
6.5 Získávání informací o OpenWrt zařízeních	45
6.6 Nasazení konfigurace	46
7 Závěr	51
Literatura	52
Seznam symbolů, veličin a zkratk	54
Seznam příloh	55
A Instalace kontroléru	56
B Adaptace nového zařízení OpenWrt	58
C Obsah příloženého CD	59

Seznam obrázků

1.1	Kanály pásma 2,4 GHz	13
1.2	SSID	14
1.3	UniFi SDN	15
1.4	OpenWISP	16
2.1	OpenWrt menuconfig	18
2.2	První přihlášení	19
2.3	Webové rozhraní LuCI	20
3.1	Princip UCI	21
3.2	Možnosti modifikace UCI	22
5.1	Role kontroléru v ukázkové topologii sítě	31
5.2	Komunikační schéma	33
5.3	Případy použití kontroléru	34
5.4	Pokus o přihlášení do kontroléru.	35
5.5	Hlavní stránka kontroléru	36
5.6	Navigační menu kontroléru	37
5.7	Stránka detailu zařízení	38
5.8	Přehled definovaných lokálních sítí v kontroléru	38
5.9	Formulář pro definování nové lokální sítě	39
5.10	Přehled definovaných bezdrátových sítí v kontroléru	39
5.11	Formulář pro definování nové bezdrátové sítě	40
5.12	Tabulka se seznamem všech asociovaných klientů	41
6.1	Diagram databázového modelu kontroléru	43
6.2	Sekvenční diagram autentizace uživatele ke kontroléru	44
6.3	Vizualizace nastavení interního switche OpenWrt	48

Seznam tabulek

1.1	Standardy IEEE 802.11 a jejich srovnání	13
6.1	Způsoby získání hlavních informací ze zařízení OpenWrt	46
6.2	Konfigurační parametry bezdrátové sítě	49

Seznam výpisů

4.1	Ukázka kostry aplikace s frameworkem Flask	26
4.2	Povolení POST metody pro přihlašovací formulář.	27
4.3	Využití objektu request k získání položky z HTTP hlavičky.	27
4.4	Renderování šablony, která zobrazí parametr jmeno.	28
4.5	Příklad šablony jazyka Jinja2.	28
4.6	Databázový model SQLAlchemy.	29
4.7	Naplnění a dotazování SQLAlchemy databáze.	29
4.8	Ukázka vykonání akce v čistém JavaScriptu.	30
4.9	Ukázka vykonání akce pomocí jQuery.	30
6.1	Kód zkoušející dostupnost portu 80.	45

Úvod

Bezdrátové sítě jsou v dnešní době velmi používané a s rostoucími nároky například na bezpečnost nebo spolehlivost jsou stále více komplexní. Z tohoto důvodu je nutné konfigurovat velké množství síťových zařízení, která jsou běžně od různých výrobců, proto se způsob jejich konfigurace často značně liší. Efektivní konfigurace složitějších bezdrátových sítí může být problém. Jedním z řešení je například automatizace správy a konfigurací, která by přinesla mnoho výhod, čímž je zejména snížení finančních nároků.

Odlišnost typů směrovačů či bezdrátových přístupových bodů v síti se dá vyřešit použitím distribuce OpenWrt [1] na těchto zařízeních. Tím budou mít všechny tyto síťové prvky jednotné rozhraní pro konfiguraci a správu, nezávisle na jejich modelu, výrobce či revizi. Distribuce OpenWrt poskytuje jednoduché rozhraní pro vzdálenou konfiguraci - JSON-RPC [6], což je množina HTTP POST požadavků, které distribuce OpenWrt implementuje [4]. Přes JSON-RPC lze například konfigurovat síť, nastavovat parametry rádia a získávat jakékoliv informace o stavu zařízení.

Cílem této práce je vytvořit kontrolér, který bude schopen vzdáleně konfigurovat a spravovat síťové prvky s distribucí OpenWrt, čehož se dá dosáhnout právě využitím rozhraní JSON-RPC.

V kapitole 1 je popsána základní teorie bezdrátových sítí dle standardů IEEE 802.11, existující řešení sloužící pro vzdálenou konfiguraci a správu bezdrátových sítí a navrhané řešení. V kapitole 2 je blíže popsána distribuce OpenWrt, způsoby její instalace na zařízení, kompilace, systém balíčkování a způsoby konfigurace. Kapitola 3 rozebírá konfigurační model distribuce OpenWrt a možnosti vzdálené konfigurace. V kapitole 4 jsou popsány použité technologie. Kapitola 5 se zabývá návrhem architektury kontroléru a návrhem uživatelského rozhraní. Kapitola 6 popisuje implementaci kontroléru.

1 Lokální bezdrátové sítě

Lokální bezdrátové sítě (WLAN) spojují dvě nebo více zařízení pomocí bezdrátové distribuční metody v omezeném prostoru [14]. Typické využití těchto sítí jsou domácnosti, školy nebo kancelářské budovy. Většina moderních sítí WLAN je založena na standardech IEEE 802.11, uváděných na trh pod značkou Wi-Fi. V kapitole 1.1 budou tyto standardy popsány podrobněji.

WLAN s větším počtem aktivních prvků (přístupových bodů, přepínačů, směrovačů) se ručně velice obtížně konfiguruje a udržuje, proto vzniklo mnoho systémů, které řeší problém centrální konfigurace a správy WLAN sítí. Tyto systémy umožňují, zpravidla přes webové rozhraní, nastavovat prvky sítě a hromadně nasazovat konfiguraci. Příkladem těchto systémů se věnuje kapitola 1.2.

1.1 Bezdrátové sítě dle standardu IEEE 802.11

IEEE 802.11 [13] je množina specifikací přístupu k médiu (MAC) a fyzické vrstvě (PHY) pro implementaci lokálních bezdrátových sítí v různých frekvencích, nejčastěji 2,4 GHz a 5 GHz. IEEE 802.11 jsou nejrozšířenější standardy bezdrátových sítí na světě. Základní verze standardu 802.11 byla vydána v roce 1997 a od té doby k ní přibyla řada rozšíření.

Třída standardů 802.11 se skládá z řady polovičně duplexních modulačních metod, které sdílí stejný základní protokol [13]. První široce akceptovaný standard z třídy bezdrátových sítí 802.11 byl 802.11b následovaný standardy 802.11a, 802.11g, 802.11n a 802.11ac. Ostatní standardy z třídy 802.11 (*c-f*, *h*, *j*) slouží jako dodatky, které rozšiřují existující standard nebo opravují nějaké jeho nedostatky. Tabulka 1.1 obsahuje bližší přehled standardů IEEE 802.11.

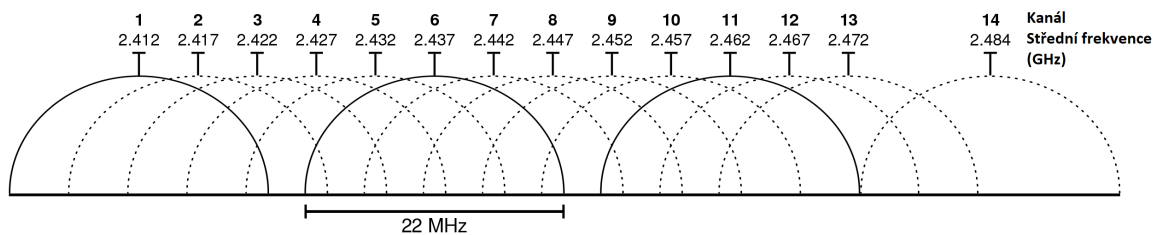
Standardy 802.11b a 802.11g využívají pásmo 2,4 GHz, z tohoto důvodu trpí zařízení v těchto sítích rušením mikrovlnnými troubami nebo zařízeními s Bluetooth. Oproti tomu standardy IEEE 802.11a a 802.11ac používají 5 GHz pásmo a nejsou tedy ovlivněny zařízeními pracujícími v pásmu 2,4 GHz.

Kanály a frekvenční pásma

Standardy 802.11b, 802.11g a 802.11n-2.4 používají 2,4 - 2,5 GHz spektrum, 802.11a a 802.11ac používají 4,915 - 5,825 GHz spektrum. Každé spektrum je ještě rozděleno do kanálů se střední frekvencí a šířkou. Pásmo 2,4 GHz je rozděleno do 14 kanálů s 5 MHz mezerami, začínající kanálem 1, který má střední frekvenci 2,412 GHz. Vizualizace pásma 2,4 GHz je zobrazena na obrázku 1.1.

Standard	Rok vydání	Pásmo [GHz]	Maximální rychlost [Mbit/s]	Fyzická vrstva
IEEE 802.11	1997	2,4	2	DSSS s FHSS
IEEE 802.11a	1999	5	54	OFDM
IEEE 802.11b	1999	2,4	11	DSSS
IEEE 802.11g	2003	2,4	54	OFDM
IEEE 802.11y	2008	3,7	54	OFDM
IEEE 802.11n	2009	2,4 nebo 5	600	MIMO OFDM
IEEE 802.11ad	2012	2,4, 5 a 60	7000	DMG
IEEE 802.11ac	2013	2,4 a 5	1000	MU-MIMO OFDM

Tab. 1.1: Standardy IEEE 802.11 a jejich srovnání.



Obr. 1.1: Vizualizace kanálů a jejich středních frekvencí v pásmu 2,4 GHz.

1.1.1 Sady služeb

Ve standardech IEEE 802.11 je sada služeb množina zařízení v bezdrátové síti pracující se stejnými síťovými parametry [14]. Množiny služeb jsou uspořádány hierarchicky.

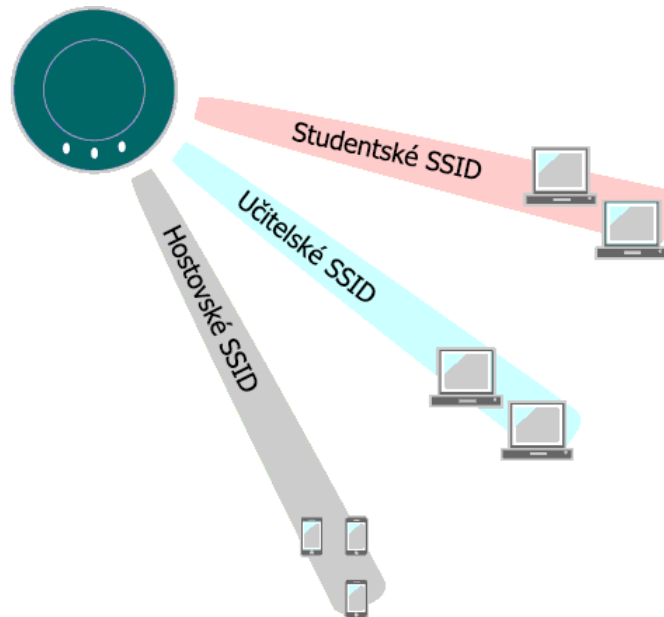
Základní sada služeb (BSS)

Základní sada služeb je množina všech zařízení, které spolu mohou komunikovat na PHY vrstvě - pracují na stejné frekvenci, se stejnou modulací apod [14]. Každý BSS má identifikační kód (ID), nazvaný BSSID, což je MAC adresa přístupového bodu obsluhující BSS. K dispozici jsou dva typy BSS: Nezávislé BSS (označovaný také jako IBSS) a infrastrukturní síť. Nezávislé BSS (IBSS) je síť ad hoc, která neobsahuje žádné přístupové body, což znamená, že se nemůže připojit k jiné základní sadě služeb.

Identifikátor sady služeb (SSID)

SSID je identifikátor logické bezdrátové sítě. Slouží zejména pro rozlišení více bezdrátových sítí pracujících ve stejném prostoru [14]. SSID je zpravidla konfigurovatelné, může mít až 32 znaků. SSID je pravidelně vysíláno přístupovými body v řídicích

rámciích (beacon frame) a obsahuje všechny informace o bezdrátové síti. Ukázka přístupového bodu vysílajícího více SSID je zobrazena na obrázku 1.2



Obr. 1.2: Přístupový bod vysílající 3 SSID.

Rozšířená sada služeb (ESS)

Rozšířená sada služeb (ESS) je sada propojených základních služeb v síti [14]. Přístupové body v ESS jsou spojeny distribučním systémem. Každý ESS má ID nazvané SSID, což je řetězec ASCII znaků dlouhý maximálně 32 znaků.

1.2 Konfigurace bezdrátových přístupových bodů

Pro správnou konfiguraci bezdrátových přístupových je potřeba provést několik kroků:

- nastavení místní sítě a její parametry,
- nastavení parametrů rádia,
- nastavení parametrů bezdrátové sítě (SSID, parametry zabezpečení).

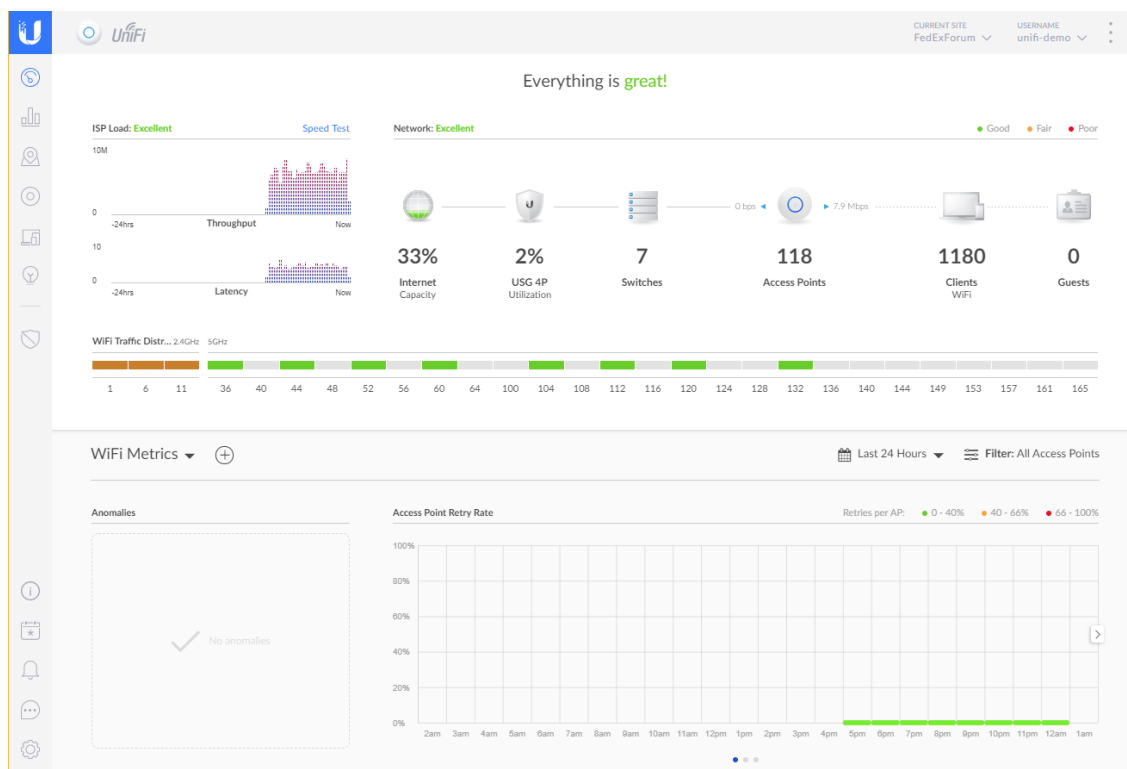
V případě, že se jedná o netriviální bezdrátovou síť s více ESSID, je potřeba také konfigurovat VLAN (Virtual Local Area Network), což je logicky nezávislá síť v rámci jednoho nebo několika zařízení. VLAN je nutné konfigurovat jak na přístupových bodech, tak i na přepínačích. Pokud je těchto přístupových bodů více,

ruční konfigurace a správa není praktické řešení. Tento problém řeší specializované nástroje na konfiguraci sítí.

1.2.1 UniFi SDN

Společnost Ubiquiti Networks nabízí ke svým síťovým zařízením webovou aplikaci UniFi [2]. UniFi slouží jako nástroj pro centralizované vytváření softwarově definovaných sítí (SDN). Z této aplikace je možné provést naprostou většinu konfigurací všech síťových prvků. Poskytuje také grafy o stavech sítí (vytíženost, propustnost), seznam síťových zařízení a jejich stav a notifikace o důležitých událostech v reálném čase. Z této aplikace je možné provést hromadné aktualizace všech nebo vybraných zařízení. Tato aplikace se prodává ve formě kompaktního zařízení, které se připojí do sítě. Ukázka grafického rozhraní aplikace UniFi je zachycena na obrázku 1.3. Demo aplikace je dostupné na webové stránce <https://demo.ubnt.com/manage/site/default/dashboard>.

Jednou z nevýhod tohoto řešení je, že funguje pouze s jinými produkty od stejného výrobce. Pro plné využití této aplikace pro správu a konfiguraci sítí je tedy nutné mít v síti pouze kompatibilní zařízení.

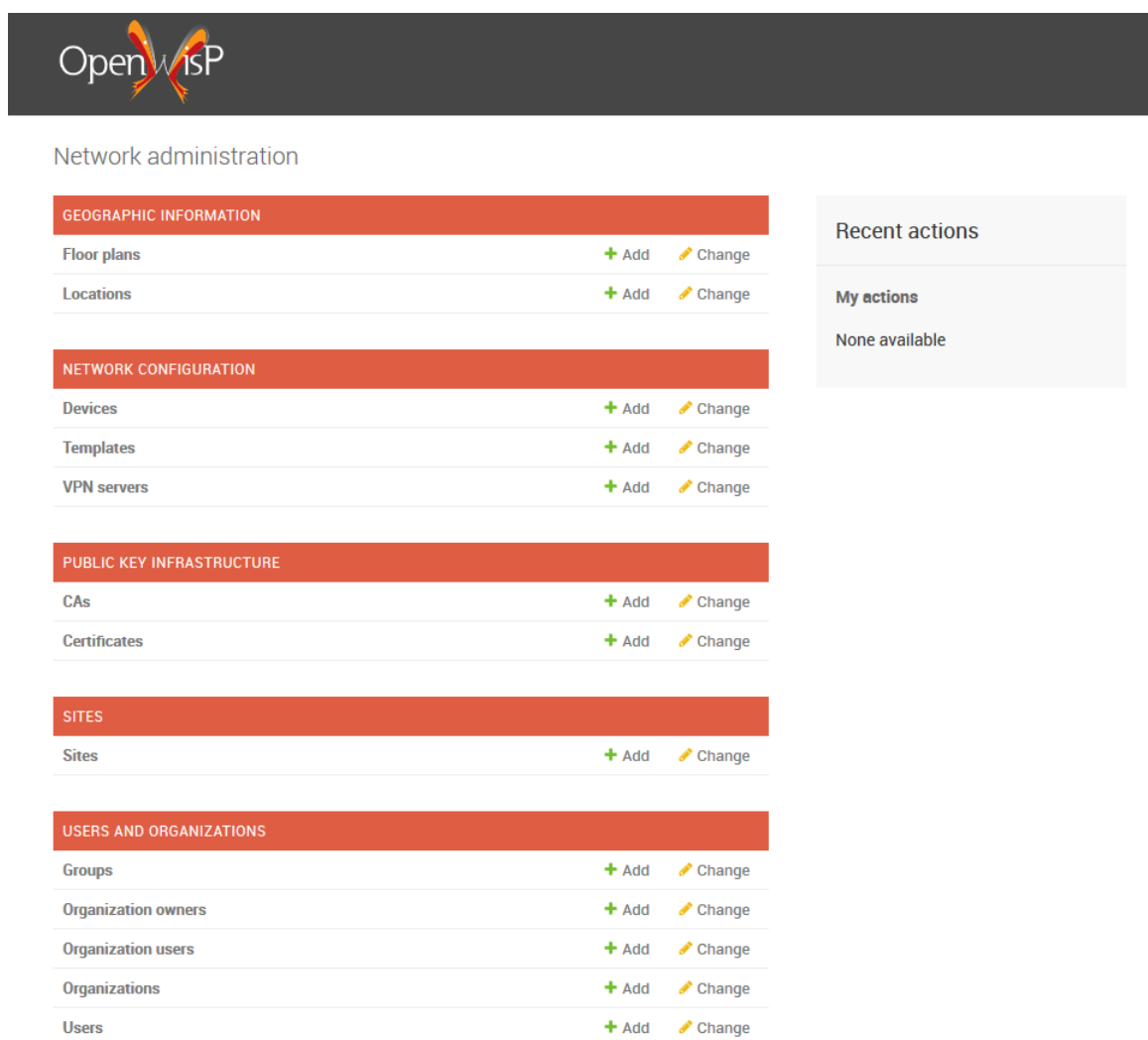


Obr. 1.3: Uživatelské rozhraní systému UniFi. Úvodní stránka webové aplikace.

1.2.2 Platforma OpenWISP

OpenWISP je softwarová platforma s otevřeným kódem navržená k usnadnění správy a konfigurace sítí, zejména bezdrátových [3]. Aktuálně pouze podporuje správu a konfiguraci distribuce OpenWrt. Projekt OpenWISP klade velký důraz na modularitu a rozšiřitelnost, existuje řada modulů, které přidávají konkrétní funkční celky do aplikace, např. modul pro autentizaci přes RADIUS.

V základu umožňuje nastavení sítí, správu zařízení, přidávání půdorysů budov a zanášení polohy jednotlivých zařízení, konfiguraci VPN serverů a ukládání nebo nahrávání konfigurace z šablon. Ukázka webového rozhraní aplikace OpenWISP je zachycena na obrázku 1.4.



The screenshot displays the OpenWISP web application interface. At the top left is the OpenWISP logo. The main content area is titled "Network administration" and is organized into several sections, each with a red header bar:

- GEOGRAPHIC INFORMATION**
 - Floor plans: + Add, Change
 - Locations: + Add, Change
- NETWORK CONFIGURATION**
 - Devices: + Add, Change
 - Templates: + Add, Change
 - VPN servers: + Add, Change
- PUBLIC KEY INFRASTRUCTURE**
 - CAs: + Add, Change
 - Certificates: + Add, Change
- SITES**
 - Sites: + Add, Change
- USERS AND ORGANIZATIONS**
 - Groups: + Add, Change
 - Organization owners: + Add, Change
 - Organization users: + Add, Change
 - Organizations: + Add, Change
 - Users: + Add, Change

On the right side, there is a sidebar with the following sections:

- Recent actions
- My actions: None available

Obr. 1.4: Uživatelské rozhraní systému OpenWISP. Úvodní stránka webové aplikace.

2 Linuxová distribuce OpenWrt

OpenWrt je GNU/Linux distribuce s otevřeným kódem určená pro vestavěné zařízení, zejména směrovače [1]. Poskytuje plnohodnotný operační systém s vlastním balíčkovacím nástrojem. Hlavními komponentami je Linuxové jádro, standardní knihovna musl nebo glibc a příkazový procesor optimalizovaný pro vestavěná zařízení BusyBox. Poskytuje mnoho možností, jak systém upravit pro specifickou potřebu či specifický hardware - umožňuje kompletní konfiguraci komponent distribuce při kompilaci instalování balíčků. Hlavní cíl vývojářů OpenWrt je podporovat co nejvíce zařízení a zajistit co největší stabilitu systému.

Projekt OpenWrt začal v roce 2004, kdy výrobce Linksys vytvořil firmware pro směrovač WRT54G založený na kódu licencovaného pod GNU/GPL [7]. Pod podmínkami této licence musel Linksys modifikovanou verzi kódu zveřejnit. Zveřejnění umožnilo nezávislým vývojářům pokračovat ve vývoji původního firmwaru. Podpora byla původně limitována jen na směrovače řady WRT54G, ale později rozšířila na mnoho jiných směrovačů a jiných zařízení od různých výrobců.

2.1 Kompilace

Zařízení se systémem OpenWrt zpravidla nemají dostatečné prostředky potřebné ke kompilaci celé distribuce či jednotlivých balíčků [8]. Je tedy nutné provést překlad na počítači s využitím kompilátoru, který generuje kód pro cílovou platformu. Tomuto procesu se říká křížová kompilace.

OpenWrt používá k tvorbě distribucí a balíčků systém založený na projektu Buildroot, což je soubor Makefilů, které uživateli umožňují konfigurovat a vytvořit distribuci pro vestavěné zařízení. Kompilaci je možno provést na systémech s operačním systémem Linux, BSD nebo OS X. Nejprve je nutné si stáhnout git repozitář s distribucí:

```
$ git clone https://github.com/openwrt/openwrt.git
```

Hlavní větev repozitáře (master) se označuje názvem bleeding edge. Na této větvi probíhá hlavní vývoj a často obsahuje experimentální změny, které nemusí být stabilní. Pro nasazení distribuce OpenWrt na produkční zařízení se doporučuje používat stabilní verze. Momentálně je poslední stabilní verze z června 2018 na větvi `openwrt-18.06`:

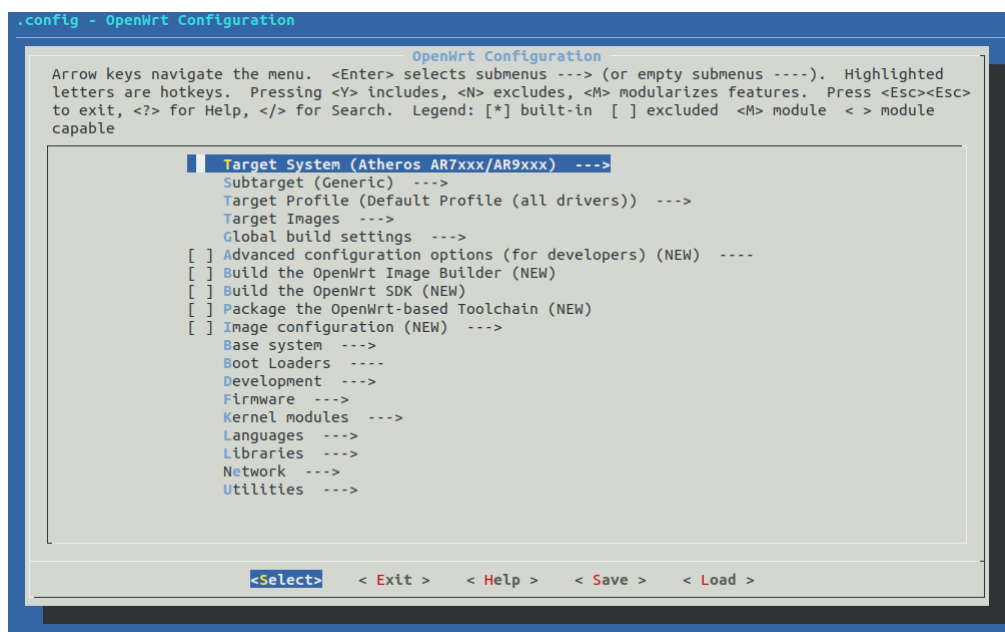
```
$ cd openwrt
$ git checkout openwrt-18.06
```

Dále je nutné stáhnout a nainstalovat dostupné balíčky:

```
$ ./scripts/feeds update -a
$ ./scripts/feeds install -a
```

Konfigurace distribuce se spouští příkazem `make menuconfig` v kořenové složce repozitáře. Po spuštění tohoto příkazu se otevře v grafické okno, ve kterém je možné specifikovat cílovou platformu, balíčky, které se zahrnou do distribuce, ovladače pro periferie atd. Menuconfig také automaticky řeší závislosti - pokud je vybraná možnost závislá na dílčích komponentech, budou automaticky zahrnuty také. Vzhled nástroje menuconfig je zachycen na obrázku 2.1.

Kompilace se spouští příkazem `make`. První kompilace trvá nejdéle, jelikož se kompiluje i samotný křížový překladač. Na běžném stroji trvá kompilace 2 hodiny. Výsledkem kompilace je soubor obrazu disku, který se nahrává na cílové zařízení.



Obr. 2.1: Úvodní obrazovka nástroje menuconfig v projektu OpenWrt. Umožňuje specifikovat nastavení parametrů kompilace a podobu výsledného obrazu disku.

2.2 Instalace

Existují čtyři nejběžnější metody instalace systému OpenWrt na zařízení [9]:

- **Přes OEM firmware** – je použito webové rozhraní původního firmwaru zařízení pro nahrání systému OpenWrt.
- **Přes bootloader a ethernetový port** – je použit protokol FTP nebo TFTP pro nahrání systému OpenWrt.

- **Přes bootloader a sériový port** – systém OpenWrt je nahrán sériovým portem.
- **Přes JTAG** – je použito rozhraní JTAG pro nahrání systému OpenWrt.

2.3 První přihlášení

Po úspěšné instalaci se může uživatel poprvé přihlásit do systému OpenWrt. Na zařízení se lze připojit pomocí protokolu Secure Shell (SSH) pod uživatelem `root` na výchozí adresu 192.168.1.1. Při prvním přihlášení je nutné, aby uživatel nastavil heslo pro uživatele `root` nástrojem `passwd`. Ukázka obrazovky při prvním přihlášení do systému OpenWrt je zachycena na obrázku 2.2.

```

BusyBox v1.25.1 () built-in shell (ash)

  LE DE  LE DE  LE DE
  DE DE  DE DE  DE DE
  LE DE  LE DE  LE DE
  DE DE  DE DE  DE DE

lede-project.org

-----
Reboot (17.01.4, r3560-79f57e422d)
-----

=== WARNING! =====
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
-----

root@LEDE:~# |

```

Obr. 2.2: Úvodní obrazovka systému OpenWrt s výzvou k nastavení hesla pro uživatele `root`.

2.4 Práce s balíčky

Stejně jako na většině Linuxových distribucí je možné využívat i na OpenWrt balíčkovací systém - `opkg` [11]. `opkg` je malý balíčkovací nástroj určený pro instalaci softwaru do vestavěných zařízení. Pro instalaci nového balíčku pomocí `opkg` je nejprve nutné si stáhnout metadata o dostupných balíčcích:

```
$ opkg update
```

Tento krok je nutné provést při každém restartu, protože se tyto informace ukládají do paměti, nikoliv na persistentní oddíl. Instalace samotného balíčku se provádí následujícím příkazem:

```
$ opkg install <NAZEV_BALICKU>
```

Pokud má instalovaný balíček závislosti na jiných balíčcích, opkg je automaticky nainstaluje také.

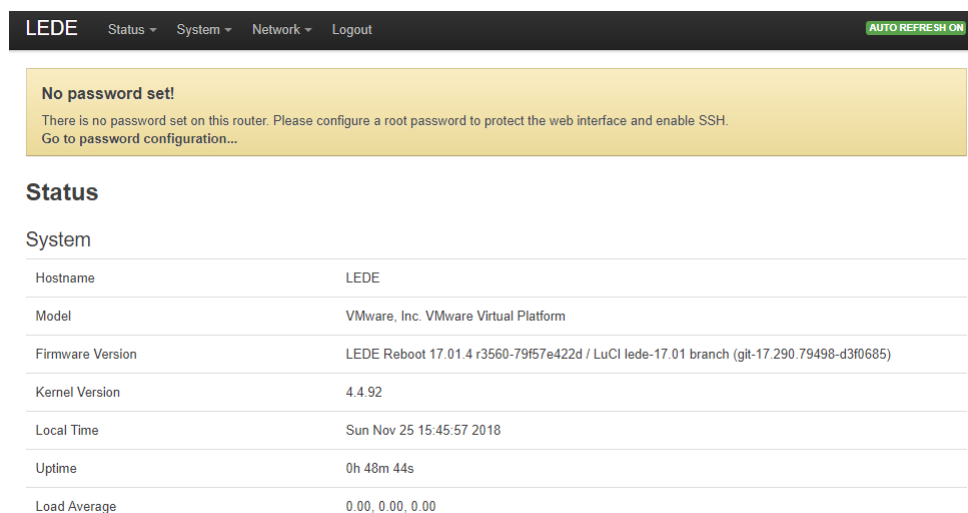
2.5 LuCI

Webové rozhraní LuCI [5] začalo být vyvíjeno v roce 2008 jako součást vydání OpenWrt verze s názvem *Kamikaze*. Důvod, proč projekt LuCI vzniknul, byla absence volného, rozšiřitelného a udržitelného webového rozhraní pro správu vestavěných zařízení. LuCI je založeno na programovacím jazyku Lua a využívá objektově orientované šablony a knihovny.

Pro použití je nutné mít nainstalovaný balík luci:

```
$ opkg update
$ opkg install luci
$ /etc/init.d/uhttpd restart
```

Nyní může uživatel využívat webové rozhraní LuCI pro konfiguraci a správu zařízení OpenWrt. Pro přihlášení je nutné použít přihlašovací údaje nějakého ze systémových účtů, např. účet `root`. Ukázka webového rozhraní LuCI je zachycena na obrázku 2.3.



Status	
System	
Hostname	LEDE
Model	VMware, Inc. VMware Virtual Platform
Firmware Version	LEDE Reboot 17.01.4 r3560-79f57e422d / LuCI lede-17.01 branch (git-17.290.79498-d3f0685)
Kernel Version	4.4.92
Local Time	Sun Nov 25 15:45:57 2018
Uptime	0h 48m 44s
Load Average	0.00, 0.00, 0.00

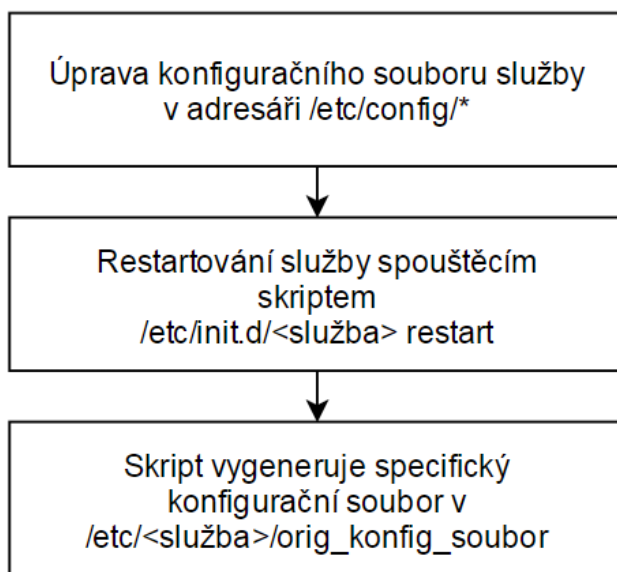
Obr. 2.3: Úvodní stránka webového rozhraní LuCI. Zobrazuje základní informace o zařízení OpenWrt.

3 Konfigurační model OpenWrt

Distribuce OpenWrt poskytuje několik způsobů, kterými lze konfigurovat zařízení. Nejpoužívanější je systém unifikovaného konfiguračního rozhraní (UCI) [12], který umožňuje konfiguraci všech základních služeb ve standardizované formě. Podrobněji je systém UCI a možnosti jeho vzdáleného použití popsán v podkapitole 3.1.

3.1 Unifikované konfigurační rozhraní

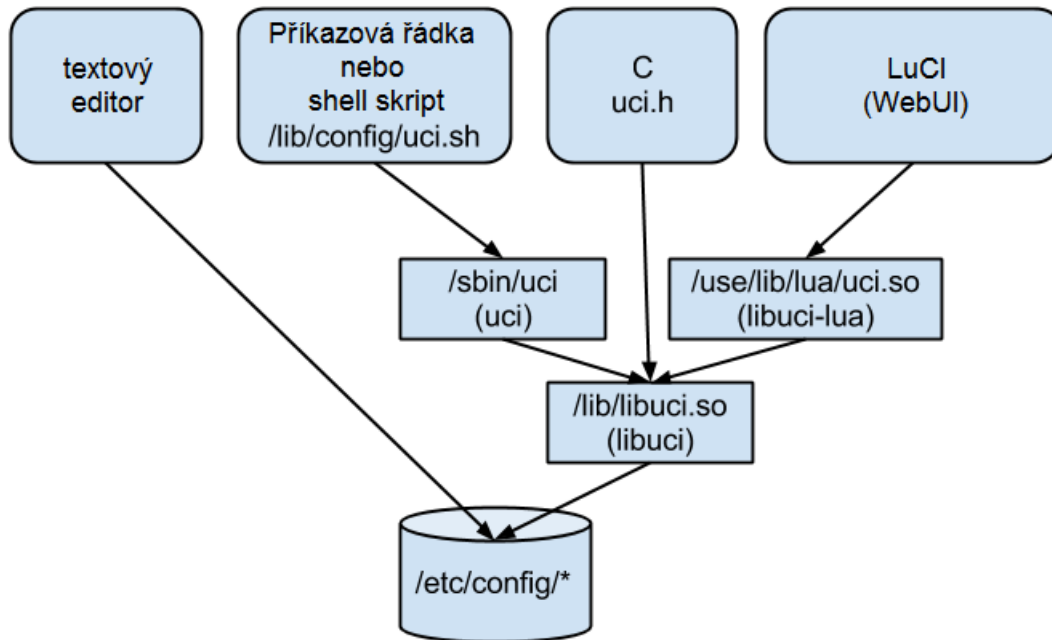
Systém UCI je založen na tvorbě meta-konfiguračních souborů se standardizovanou syntaxí v adresáři `/etc/config` [12]. Tyto soubory jsou při startu konkrétní služby typicky převedeny do formy konfiguračních souborů specifických pro jednotlivé aplikace. Daný přístup řeší problém odlišných verzí služeb a aplikací mezi systémy OpenWrt – i když je v síti více zařízení s jinou verzí, konfigurační model UCI zůstává zpravidla stejný. Obecný proces úpravy a aplikace konfiguračních souborů UCI znázorňuje schéma 3.1.



Obr. 3.1: Modifikované konfigurační soubory UCI se při restartu služby převedou do konfiguračního souboru specifické služby.

Konfigurační soubory je možné ručně editovat běžným textovým editorem nebo s nimi lze pracovat pomocí utility `uci`. Konfigurační soubory lze také upravovat pomocí několika různých programových rozhraní, jsou podporované skriptovací jazyky Lua a Shell a dále kompilovaný jazyk C [12]. Webové konfigurační rozhraní

LuCI využívá právě jazyk Lua pro modifikaci UCI souborů [5]. Schéma 3.2 zobrazuje závislosti přístupů, kterými je možné upravovat konfigurační soubory UCI, na nástrojích či knihovnách.



Obr. 3.2: Přístupy, kterými je možno manipulovat s konfiguračními soubory UCI a jejich závislosti na nástrojích či knihovnách.

3.1.1 Syntaxe

Konfigurační soubor UCI je složen z jedné nebo více `config` sekcí, které obsahují jeden nebo více `option` řádků, které definují konkrétní hodnoty.

Komentáře (řádky, které nejsou zpracovávány) jsou zapisovány znakem `#` [12]. Pokud tímto znakem začíná řádek, je celý ignorován, jinak je ignorována jen část řádku, od kterého znak začíná.

Níže je ukázka konfigurační sekce UCI:

```

config 'ukazka' 'jmeno_sekce'
    option nazev_volby 'volba1'
  
```

- Řádek `config 'ukazka' 'jmeno_sekce'` definuje začátek sekce typu `ukazka` se jménem `jmeno_sekce`. Existují také anonymní sekce - sekce bez jména obsahující pouze typ sekce.
- Řádek `option nazev_volby 'volba1'` název volby a její hodnotu v sekci.

- Odsazení jaké je použito v příkladu není potřeba dodržovat, ale zvyšuje čitelnost konfiguračního souboru.

3.1.2 Konfigurace nástrojem uci

Pro jednoduché úpravy je možné upravovat konfigurační souboru napřímo textovým editorem. Pro automatizovanou správu je vhodné použít aplikaci `uci`. Tento nástroj umožňuje veškerou manipulaci s konfiguračními soubory UCI, poskytuje také kontrolu uživatelských chyb.

Aplikaci `uci` funguje režimem transakcí, což znamená, že všechny modifikace provedené na konfiguračních objektech jsou provedeny až po explicitním aplikování [12]. Tento proces je vysvětlen na příkladu níže:

```
1 root@OpenWrt:~# uci set uhttpd.main.listen_http=8080
2 root@OpenWrt:~# uci commit uhttpd
```

- Na řádku 1 se přes aplikaci `uci` změnil port, na kterém webový server `uhttpd` poslouchá na `http` komunikaci. Po provedení tohoto řádku se konfigurační soubor `/etc/config/uhttpd/uhttpd.conf` nezměnil.
- Na řádku 2 se explicitně zapsaly provedené změny konfigurace služby `uhttpd`.

Tímto principem lze vytvořit mnoho změn v rámci jedné transakce.

3.1.3 Konfigurace přes rozhraní JSON-RPC

Projekt LuCI poskytuje některé z jeho knihoven k použití vzdáleně externími aplikacemi přes rozhraní JSON-RPC.

Rozhraní JSON-RPC

JSON-RPC je volání vzdálené procedury (RPC) vložené do javascriptové objektové notace (JSON) [6]. Tento protokol funguje na základě komunikace klient-server, kde požadavek na server i odpověď serveru klientovi jsou v JSON formátu, které obsahují protokolem definované parametry.

JSON objekt, který posílá klient, musí obsahovat atributy:

- **method** – textový řetězec obsahující jméno volané metody,
- **params** – pole parametrů,
- **id** – hodnota jakéhokoliv typu, která identifikuje požadavek.

JSON objekt, kterým odpovídá server, musí obsahovat atributy:

- **result** – data vrácená metodou. Pokud nastala chyba, musí být hodnota `null`.

- **error** – pokud nastala chyba, tak její číslo, jinak **null**,
- **id** – hodnota, která byla v klientském požadavku.

Rozhraní JSON-RPC na distribuci OpenWrt

V základní distribuci OpenWrt není JSON-RPC nainstalované, je potřeba si nainstalovat balíček `luci-mod-rpc` a následně restartovat webový server [4]:

```
$ opkg update
$ opkg install luci-mod-rpc
$ /etc/init.d/uhttpd restart
```

Po instalaci je možné využívat JSON-RPC rozhraní pro správu a konfiguraci OpenWrt z externích zařízení. Požadavky se zasílají HTTP metodou POST na rpc koncový bod rozhraní OpenWrt: `<OPENWRT_IP>/cgi-bin/luci/rpc/<MODUL>`, kde `<MODUL>` je název knihovny rozhraní. OpenWrt poskytuje přes JSON-RPC následující knihovny [4]:

- **uci** – umožňuje modifikovat UCI konfigurační soubory,
- **fs** – poskytuje rozhraní do souborového systému OpenWrt,
- **sys** – umožňuje interagovat s operačním systémem OpenWrt,
- **ipkg** – poskytuje rozhraní do balíčkovacího systému (IPKG nebo OPKG).

Pro volání většiny metod v rozhraní je potřeba provést autentizaci, kde OpenWrt dává k dispozici `login` metodu v JSON-RPC knihovně `auth` [4]. Invokace této metody s parametry uživatelského jména a hesla následně vrátí OpenWrt token, který je použit pro volání metod vyžadující autentizaci.

JSON-RPC knihovna `uci` poskytuje téměř stejnou funkcionalitu, jako aplikace `uci` z kapitoly 3.1.2, lze tedy plně konfigurovat zařízení s distribucí OpenWrt externí službou. Příklad, kterým lze modifikovat UCI konfigurační možnost služby `uhttpd`:

```
$ curl http://<OPENWRT_IP>/cgi-bin/luci/rpc/uci?auth=<TOKEN> --data '
{
  "id": 1,
  "method": "set",
  "params": [
    "uhttpd",
    "main",
    "listen_http",
    "8080"
  ]
}'
```

Obdobným způsobem lze přidávat celé sekce, mazat či měnit konfigurační možnosti. Všechny změny v UCI se i přes JSON-RPC chovají transakčně, pro aplikování změn je tedy nutné zavolat metodu `commit` [4]. Přehled JSON-RPC metod dostupných v knihovně `uci` je uveden v dokumentaci [4].

4 Použité technologie

Existuje mnoho frameworků k tvorbě webových aplikací. Jelikož kontrolér je v principu jednoduchá webová aplikace, byl pro jeho implementaci zvolen lehký Python framework Flask [15]. Pro tvorbu moderního a responzivního vzhledu aplikace byla vybrána knihovna Bootstrap 3, který má přes modul podporu ve frameworku Flask. V následujících kapitolách jsou použité technologie podrobněji popsány.

4.1 Webový aplikační framework Flask

Flask je lehký webový aplikační framework napsaný v programovacím jazyce Python. Jelikož je jeho jádro velice jednoduché, nazývá se někdy jako mikroframework. Flask je založen na jednoduché rozšiřitelnosti [15], bylo pro něj vyvinuto mnoho modulů implementující řešení specifických problémů, jako databáze nebo validace formulářů. Kostra fungující Flask aplikace, uvedená ve výpisu 4.1, vyžaduje pouze 9 řádků.

```
1 from flask import Flask
2 app = Flask(__name__)
3
4 @app.route('/')
5 def ahoj_sвете():
6     return 'Ahoj_Svete!'
7
8 if __name__ == '__main__':
9     app.run()
```

Výpis 4.1: Ukázka kostry aplikace s frameworkem Flask

Flask je postaven na Web Server Gateway Interface (WSGI), což je specifikace pro univerzální rozhraní mezi webovými aplikacemi a webovými servery. Vývoj a nasazení aplikací je velice jednoduché, protože Flask obsahuje vlastní vývojový server, který je součástí instalace. Pro generování HTML šablon využívá Flask šablonovací jazyk Jinja2.

Hlavní důvody pro volbu frameworku Flask:

- **Rychlost vývoje** – Vývoj jednoduché webové aplikace je prakticky vždy rychlejší s minimalistickým frameworkem Flask oproti např. frameworku Django.
- **Učící křivka** – Flask má velice snadnou konfiguraci a plnou dokumentaci v kódu pomocí nástroje Docstrings.
- **Multiplatformí podpora** – Flask může běžet na všech moderních operačních systémech.

- **Licence** – Flask má otevřený kód pod licenci BSD.

Routování

Python dekorátor `@app.route` je použit pro svázání funkce s URL [15]. Parametr použitý v dekorátoru svazuje funkci k relativní cestě. Příklad cesty je `'/'` pro kořen stránky a `'/ahoj_svete'` pro vnořenou stránku. Příklad uvedený v 4.1 svazuje funkci `ahoj_svete()` k URL `hostname/`.

HTTP metody

Nepovinným parametrem pro `@app.route` dekorátor je pole povolených HTTP metod. Pokud není tento parametr specifikován, je vždy povolena HTTP metoda `GET` [15]. Příklad povolení HTTP metody `POST` je uveden ve výpisu 4.2.

```
1 @app.route('/login', methods=['GET', 'POST'])
2 def login():
3     if request.method == 'POST':
4         prihlas_uzivatele()
5     else:
6         zobraz_prihlasovaci_formular()
```

Výpis 4.2: Povolení `POST` metody pro přihlašovací formulář.

Požadavky

Práce s příchozími požadavky se děje přes globální objekt `request` [15]. Tento objekt obsahuje všechny atributy příchozího požadavku jako řetězec dotazu, HTTP hlavičky, formulářová data nebo požadavek `POST` a nahrané soubory. Výpis 4.3 ukazuje jednoduché použití objektu `request` pro získání položky `'User-Agent'` z HTTP hlavičky.

```
1 @app.route('/user_agent')
2 def user_agent():
3     user_agent_name = request.headers.get('User-Agent')
4     return user_agent_name
```

Výpis 4.3: Využití objektu `request` k získání položky z HTTP hlavičky.

Odpovědi

Při navracení odpovědi, která bude renderována jako webová stránka, Flask poskytuje několik možností, které tento proces dělají flexibilní. Například vrácený řetězec

bude automaticky převeden na validní HTML stránku. Existuje několik funkcí pro vytváření specifických odpovědí, například `send_file(cesta, souboru)`, která uživateli pošle soubor. HTML soubory mohou být renderovány funkcí `make_response()`, složitější stránky můžou být vytvořeny s pomocí šablonovacího jazyka. Šablonovací jazyk použitý ve frameworku Flask je vysvětlen v následující sekci.

Šablony

Flask používá šablonovací jazyk zvaný Jinja2, který značně pomáhá vývojáři k zajištění bezpečnosti aplikace. Šablonovací jazyk automaticky ukončuje potenciálně nebezpečné uživatelské vstupy, aniž by na to musel vývojář myslet [15].

```
1 from flask import render_template
2
3 @app.route('/<jmeno>')
4 def ahoj(jmeno=None):
5     return render_template('ahoj.html', jmeno=jmeno)
```

Výpis 4.4: Renderování šablony, která zobrazí parametr jmeno.

```
1 <!doctype html>
2 <title>Flask Jinja2 priklad</title>
3
4 <h1>Ahoj {{ jmeno }}!</h1>
```

Výpis 4.5: Příklad šablony jazyka Jinja2.

Výpis 4.4 zobrazuje jednoduchý příklad renderování HTML stránky pomocí šablony. Funkce `render_template` načte šablonu uvedenou ve výpisu 4.5 a dosadí za proměnnou `jmeno` řetězcem zadaným v URL. Šablonovací jazyk se postará o všechna ukončení uživatelských vstupů, takže potenciální útočník nemůže poškodit stránku přes URL.

SQLAlchemy

SQLAlchemy je knihovna určená pro práci nad SQL databázemi přímo z prostředí jazyka Python [19]. Cílem knihovny je poskytnout efektivní a výkonný databázový přístup. SQLAlchemy podporuje všechny běžné poskytovatele databáze, např. MySQL, PostgreSQL nebo SQLite, což je databáze držená v paměti programu.

Kód zobrazený ve výpisu 4.6 ukazuje, jak se vytváří spojení k databázi. Databázové URI je knihovně předáno přes konfigurační atribut Flask aplikace a následně

je vytvořen databázový objekt. Tento objekt obsahuje třídu `Model`, která může být použita k deklaraci databázového modelu - databázové tabulky. V tomto příkladu je vytvořen model `Uzivatel`.

```
1 from flask import Flask
2 from flask_sqlalchemy import SQLAlchemy
3
4 app = Flask(__name__)
5 app.config['SQLALCHEMY_DATABASE_URI'] = \
6     'sqlite:///tmp/db.sqlite'
7 db = SQLAlchemy(app)
8
9 class Uzivatel(db.Model)
10     id = db.Column(db.Integer, primary_key=True)
11     jmeno = db.Column(db.String(64), unique=True)
12     email = db.Column(db.String(128), unique=True)
13
14     def __init__(self, jmeno, email)
15         self.jmeno = jmeno
16         self.email = email
```

Výpis 4.6: Databázový model SQLAlchemy.

Po vytvoření databázového modelu se může databáze naplnit konkrétními záznamy. Výpis 4.7 zobrazuje práci s SQLAlchemy databází. Mezi řádky 1 až 6 jsou uživatelé vytvořeni a vloženi do databáze. Následně je databáze dotazována. Nejprve je dotázána na všechny uživatele, poté je dotázána pouze na prvního uživatele se jménem 'Host'.

```
1 admin = Uzivatel('admin', 'admin@admin.cz')
2 host = Uzivatel('host', 'host@host.cz')
3
4 db.session.add(admin)
5 db.session.add(host)
6 db.session.commit()
7
8 uzivatele = Uzivatel.query.all()
9 host = Uzivatel.query.filter_by(jmeno='host').first()
```

Výpis 4.7: Naplnění a dotazování SQLAlchemy databáze.

4.2 JavaScriptová knihovna jQuery

jQuery je JavaScriptová knihovna běžící na straně klienta (prohlížeče), která slouží k manipulaci objektové reprezentace HTML dokumentu (DOM) [17]. Důvodem použití jQuery oproti čistému JavaScriptu je podstatné zjednodušení práce a zpřehlednění zdrojového kódu. Dále jQuery abstrahuje typ prohlížeče, ve kterém běží, takže vývojář automaticky píše kód, který bude fungovat ve všech podporovaných prohlížečích.

jQuery poskytuje následující funkcionalitu:

- Manipulaci s HTML/DOM,
- manipulaci s CSS,
- registrace na události vyvolané v dokumentu,
- efekty a animace,
- AJAX a pomocné funkce.

Pro ilustraci síly a důvodu širokého použití jQuery, výpis 4.8 napsaný v klasičtém JavaScriptu na 5 řádků vykonává stejnou akci jako výpis 4.9 napsaný pomocí knihovny jQuery na 1 rádek.

```
1 var d = document.getElementsByClassName("element");
2 var i;
3 for (i = 0; i < d.length; i++) {
4     d[i].className = d[i].className + " selected";
5 }
```

Výpis 4.8: Ukázka vykonání akce v čistém JavaScriptu.

```
1 $(".element").addClass("selected");
```

Výpis 4.9: Ukázka vykonání akce pomocí jQuery.

5 Návrh řešení

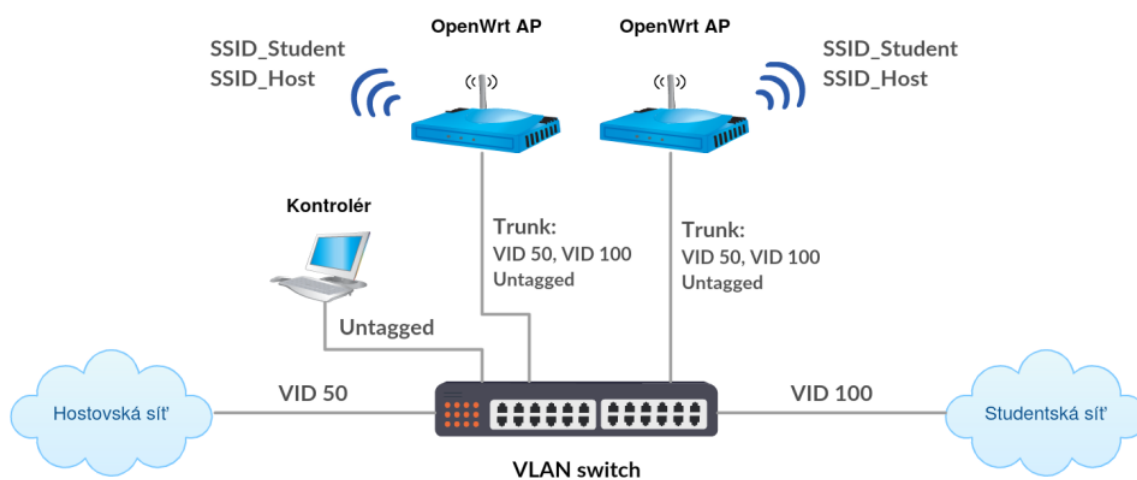
V této kapitole bude popsán návrh kontroléru, který bude sloužit pro centralizovanou konfiguraci a správu přístupových bodů v sítích IEEE 802.11 se systémy OpenWrt.

Kontrolér je koncipovaný jako webová aplikace postavená na frameworku Flask [15], který je napsán v programovacím jazyku Python [16]. Ke komunikaci s jednotlivými přístupovými body se systémem OpenWrt bylo zvoleno rozhraní JSON-RPC, které je popsáno v kapitole 3.1.3.

5.1 Role kontroléru

Kontrolér má roli centralizovaného nástroje, který sbírá informace o jednotlivých zařízeních s OpenWrt systémy a dokáže je konfigurovat. Kontrolér tedy musí být ve stejné síti, jako OpenWrt zařízení, aby s nimi mohl komunikovat. Pro přístup na kontrolér stačí jakýkoliv webový prohlížeč.

Příklad síťové topologie, do které je zapojený kontrolér a dva přístupové body OpenWrt, je znázorněn na obrázku 5.1.



Obr. 5.1: Role kontroléru v ukázkové topologii sítě. Kontrolér konfiguruje přístupové body OpenWrt, aby vytvořily přístup ke studentské a hostovské síti přes bezdrátové sítě SSID_Student a SSID_Host.

V rámci ukázkové topologie se nejprve nakonfigurují jednotlivé porty na přepínači:

- Portu, do kterého je připojena hostovská síť, se přiřadí tag VID 50.

- Portu, do kterého je připojena studentská síť, se přiřadí tag VID 100.
- Porty, do kterých jsou zapojeny zařízení OpenWrt, se nakonfigurují jako trunk.

Poté se na webovém rozhraní kontroléru nakonfigurují bezdrátové sítě, jejich parametry a přiřadí se k lokálním sítím, ke kterým budou koncoví klienti připojení:

- Paketům, které přijdou z bezdrátové hostovské sítě `SSID_Host`, se přiřadí VLAN tag 50, tím budou klienti schopni komunikovat se zařízeními v hostovské síti.
- Paketům, které přijdou z bezdrátové hostovské sítě `SSID_Student`, se přiřadí VLAN tag 100, tím budou klienti schopni komunikovat se zařízeními ve studentské síti.

Nakonec se tato nadefinovaná konfigurace nasadí na OpenWrt zařízení. Způsob nasazení konfigurace je popsán níže.

Způsob nasazení konfigurace

Kontrolér komunikuje s OpenWrt zařízeními přes rozhraní JSON-RPC, které je popsáno v sekci 3.1.3. Princip konfigurace je znázorněn sekvenčním diagramem na schématu 5.2. Nejprve je nutné na provést na webovém rozhraní kontroléru konfiguraci lokálních a bezdrátových sítí, poté se tato konfigurace může nasadit na jednotlivá OpenWrt zařízení. Dále je nutné získat autentizační token pro rozhraní JSON-RPC. Poté se vymaže stará konfigurace na OpenWrt zařízení, nahraje aktuální a aplikuje se potvrzením transakce. Nakonec dá webové rozhraní kontroléru uživateli zpětnou vazbu o výsledku provedení nasazení konfigurace.

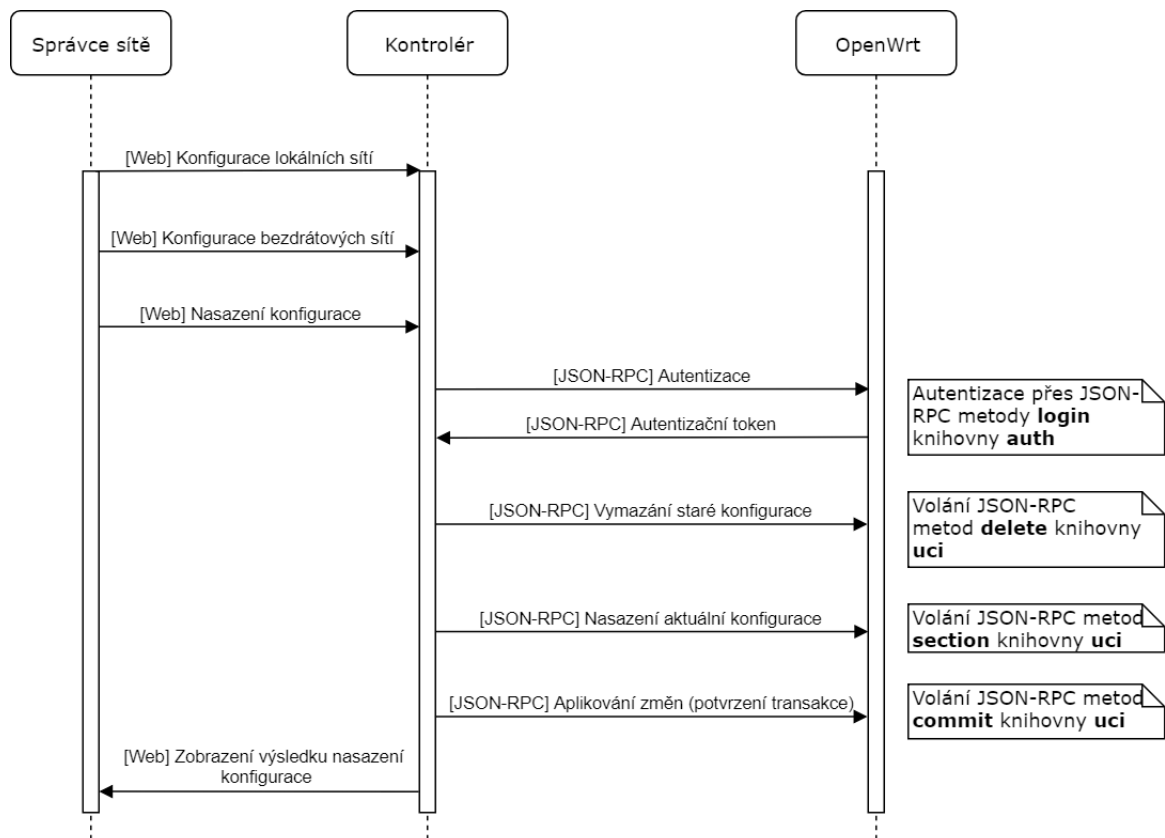
5.2 Formulace požadavků

Kontrolér musí umožnit uživateli zejména vytvořit ucelenou síť, nakonfigurovat bezdrátové sítě a získat základní přehled o stavu jednotlivých přístupových bodů. Při přidání nového přístupového bodu do sítě musí kontrolér umožnit jeho jednoduchou adaptaci. Na obrázku 5.3 je popsán diagram případů použití z pohledu správce sítě.

5.2.1 Definování bezdrátových sítí

Bezdrátová síť je definována následujícími parametry:

- Identifikátorem bezdrátové sítě (SSID),
- způsobem zabezpečení,
- lokální sítí, se kterou je v bridge módu,
- možností skrýt SSID,



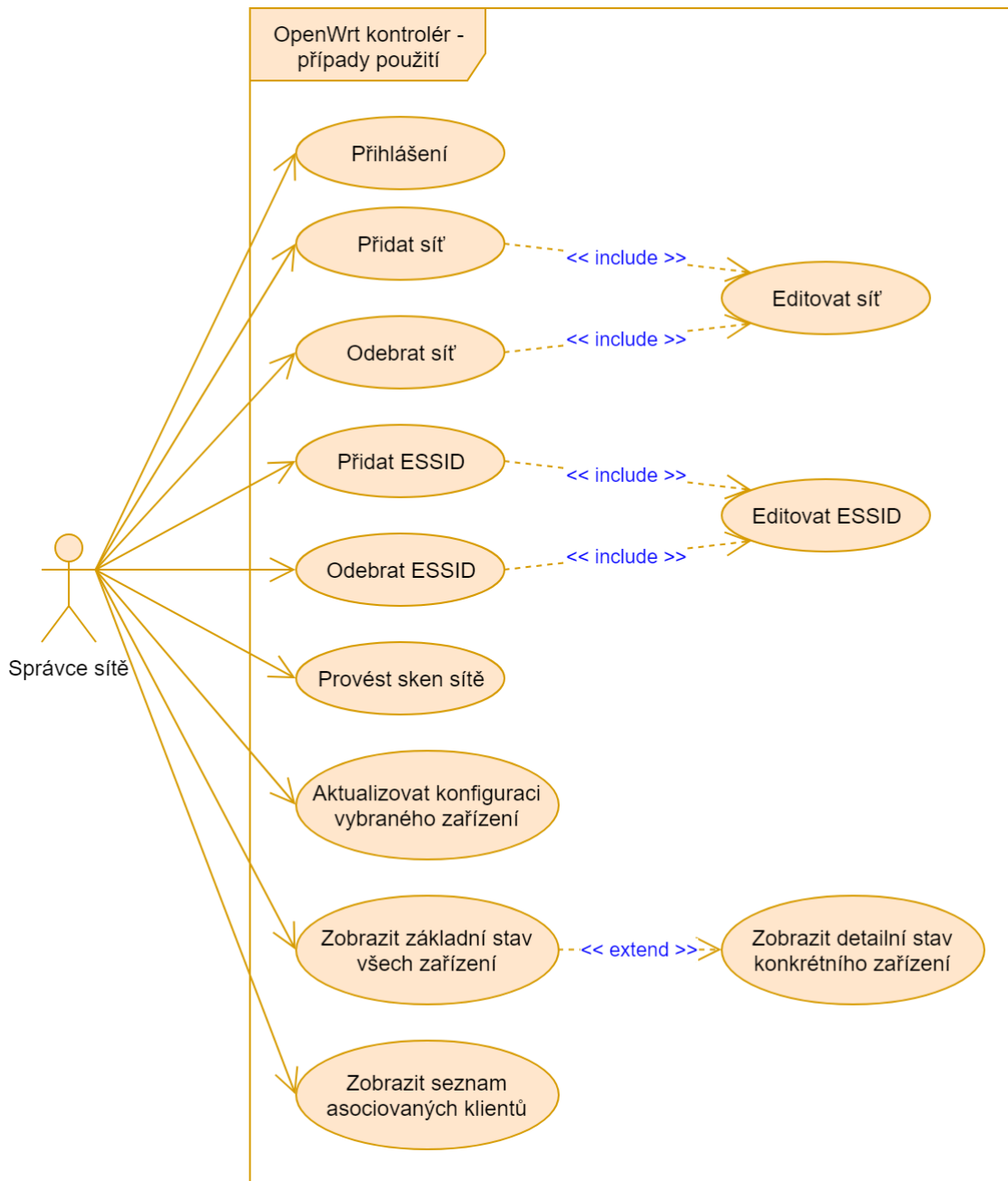
Obr. 5.2: Sekvenční diagram znázorňující nasazení konfigurace na jednotlivé přístupové body se systémem OpenWrt.

- možností izolování klientů v bezdrátové síti.

5.2.2 Zobrazení stavu zařízení

Kontrolér musí mít přehled o stavu sledovaných přístupových bodů, zejména o:

- Názvu zařízení,
- dostupnosti zařízení (dostupné / nedostupné),
- stavu rozhraní JSON-RPC (dostupné / nedostupné),
- verzi systému,
- hardwarové revizi zařízení,
- době běhu systému,
- aktuálním zatížením zařízení,
- aktuálním stavu paměti zařízení,
- aktuálním nastavením lokálních sítí,
- aktuálním nastavením bezdrátových sítí,
- systémovém logu,



Obr. 5.3: UML Diagram případů použití kontroléru z pohledu uživatele (správce sítě).

- seznamu asociovaných klientů v bezdrátových sítích.

5.3 Návrh uživatelského rozhraní

Uživatelské rozhraní musí uživateli umožnit vykonávání všech funkcí, které jsou popsány v diagramu případu použití 5.3. Jednotlivé části uživatelského rozhraní popisují obrázky v následujících podkapitolách, které blíže popisuje doprovodný text.

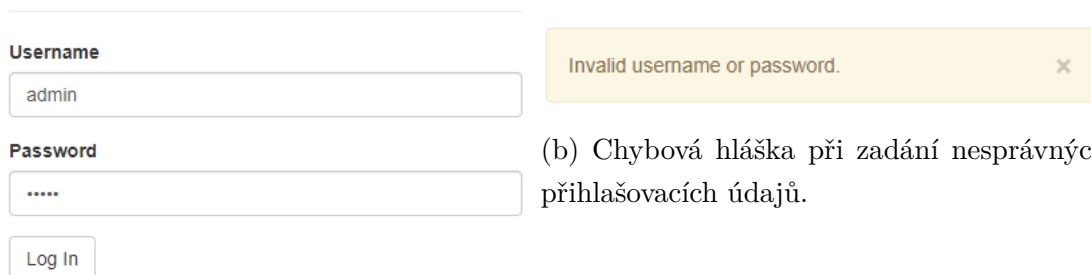
5.3.1 Přihlašovací stránka

Při prvním spuštění aplikace je uživatel vyzván k přihlášení na stránce `/login`. Pro přihlášení musí uživatel zadat platné uživatelské jméno a heslo v přihlašovacím formuláři zobrazeném na obrázku 5.4a. Pokud je zadána neplatná kombinace uživatelského jména a hesla, zobrazí se uživateli informační zpráva 5.4b o této události.

Pokud se uživatel pokusí zobrazit přihlašovací stránku a je již přihlášen, bude automaticky přesměrován na hlavní stránku.

Authorization Required

Please enter your username and password.



Username

Password

Log In

Invalid username or password. ✕

(a) Přihlašovací okno kontroléru.

(b) Chybová hláška při zadání nesprávných přihlašovacích údajů.

Obr. 5.4: Pokus o přihlášení do kontroléru.

5.3.2 Hlavní stránka

Po úspěšném přihlášení je uživatel přesměrován na hlavní stránku. Účelem hlavní stránky je zobrazení přehledu dostupných OpenWrt zařízení ve sledované síti, zobrazení informací o jejich stavu a možnost vykonávat akce nad jednotlivými zařízeními. Obrázek 5.5 ilustruje vzhled hlavní stránky.

Hlavním elementem stránky je tabulka, kde každý její řádek je detekované OpenWrt zařízení ve spravované síti, která je uvedena v nadpisu nad tabulkou. V samotné tabulce jsou obsaženy základní informace o zařízení, ze kterých uživatel může

na první pohled zjistit, zda se zařízení nachází v dobrém nebo nežádoucím stavu. Dále obsahuje tabulka prvky umožňující provádění změn na zařízení.

Access Points (192.168.1.0/24)

Name	IP Address	Ping	LuCI	SSH	Hostname	Firmware	Uptime	Clients	Comment	WiFi channel	Actions
OpenWRT1	192.168.1.1	✓	✓	✓	OpenWrt	18.06.1	37d 6h	0	Mistnost C203	⌵	Update

[Scan Network](#)

Scan finished ✓ (16:59:03 30.04.2019)

1/1

Obr. 5.5: Hlavní stránka kontroléru.

Tabulka seznamu zařízení obsahuje následující sloupce:

- **Name** – obsahuje tlačítko s vygenerovaným unikátním jménem pro zařízení, po jehož zmáčknutí se zobrazí stránka s detailními informacemi o zařízení popsaná v kapitole 5.3.3.
- **IP Address** – odkaz s IP adresou, na které je zařízení dostupné.
- **Ping** – indikuje, zda je zařízení dostupné na síti.
- **LuCI** – indikuje, zda je na zařízení dostupné rozhraní JSON-RPC.
- **SSH** – indikuje, zda je zařízení dostupné přes protokol Secure Shell (SSH).
- **Hostname** – trvalé jméno zařízení.
- **Uptime** – celková doba provozu zařízení od jeho posledního spuštění či restartu.
- **Clients** – počet wifi klientů připojených přes zařízení.
- **Comment** – komentář, který může uživatel ke konkrétnímu zařízení nastavit pro jeho bližší identifikaci.
- **WiFi channel** – WiFi kanál, na kterém zařízení vysílá. Je možné nastavit konkrétní kanál nebo nastavit automatický mód, kdy se zařízení pokusí zvolit nejvhodnější kanál.
- **Actions** – tlačítko, které po jeho kliknutí nahraje aktuální konfiguraci definovanou na kontroléru na zařízení.

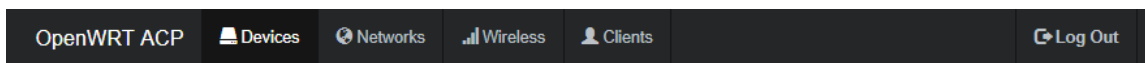
Pod tabulkou se nachází sekce s informacemi o probíhajícím či posledním vyhledávání dostupných OpenWrt zařízení ve sledované síti. Pokud aktuálně sken sítě neprobíhá, je zobrazeno datum a čas posledního vykonaného skenu. Uživatel tak ví, jestli jsou informace o zobrazených zařízeních v tabulce aktuální. Při zmáčknutí

tlačítka **Scan Network** se spustí procedura hledání dostupných zařízení a uživatel je o jejím stavu informován grafickým indikátorem zobrazující průběh skenování.

Navigační menu

Pokud je uživatel přihlášen, zobrazí se mu na všech stránkách navigační menu v horní části obrazovky znázorněné na obrázku 5.6.

- Nápis **OpenWRT ACP** a tlačítka **Devices** směřují na domovskou stránku pod adresou '/',
- tlačítka **Networks** směřuje na stránku s definicí lokálních sítí pod adresou '/network',
- tlačítka **Wireless** směřuje na stránku s definicí bezdrátových sítí pod adresou '/wireless',
- tlačítka **Clients** směřuje na stránku s tabulkou obsahující všechny připojené klienty pod adresou '/clients',
- tlačítka **Log Out** odhlásí uživatele a přesměruje na přihlašovací stránku pod adresou '/login'.



Obr. 5.6: Navigační menu kontroléru.

5.3.3 Detail zařízení

Stránka s detailem zařízení 5.7 obsahuje množství bližších informací o konkrétním zařízení. Slouží zejména pro stavy, kdy zařízení nepracuje správně. Z informací obsažených na této stránce dokáže uživatel s velkou pravděpodobností určit příčinu, proč je zařízení v nežádoucím stavu.

Stránka obsahuje pět záložek:

- **Device** – informace o systému, zejména o typu hardwarového modelu zařízení, míry zatížení a stavu paměti.
- **Network** – výpis aktuální konfigurace síťových rozhraní.
- **Wireless** – výpis aktuální konfigurace bezdrátových rozhraní.
- **Syslog** – systémový log zařízení.
- **Kernel Log** – log linuxového jádra zařízení.

OpenWRT1

Device Network Wireless Syslog Kernel Log

System

Hostname	OpenWrt
Model	TP-Link TL-WR1043N/ND v4
Firmware Version	OpenWrt 18.06.1 r7258-5eb055306f
Kernel Version	4.9.120
Local Time	2019-03-23 12:15:00
Uptime	38d 6h 21m
Load Average	0.01 0.03 0.00 1/52 6397

Memory

Free (35004 kB / 60320 kB) Used (25316 kB / 60320 kB)

Obr. 5.7: Stránka detailních informací o zařízení.

5.3.4 Definice lokálních sítí

Seznam definovaných lokálních sítí se nechází na stránce 5.8 dostupné pod adresou '/network'. V tabulce jsou zobrazeny již nadefinované sítě, které jdou upravit nebo mazat tlačítka ve sloupci **Actions**. Definovat novou síť lze kliknutím na tlačítko + **Add Network**.

Name	VLAN	Configure Gateway	Gateway IP	Network Address	Actions
network_1	10	False	-	-	Edit Delete
network_2	20	False	-	-	Edit Delete

[+ Add Network](#)

Obr. 5.8: Přehled definovaných lokálních sítí v kontroléru.

Stránka 5.9 s editací nebo definováním nové sítě obsahuje formulář, ve kterém musí uživatel vyplnit následující textová pole:

- **Name** – unikátní název sítě v rámci kontroléru,
- **VLAN** – číslo VLAN identifikátoru, musí být unikátní v rámci kontroléru.

Name
VLAN
Configure Gateway

Obr. 5.9: Formulář pro definování nové lokální sítě v kontroléru.

Volitelně může uživatel nadefinovat výchozí bránu sítě. Tato možnost má využití zejména v případě, že v definované síti neexistuje žádný DHCP server.

5.3.5 Definice bezdrátových sítí

Seznam definovaných bezdrátových sítí se nachází na stránce 5.10 dostupné pod adresou `/wireless`. Podobně jako u stránky s přehledem lokálních sítí je zde přehled definovaných bezdrátových sítí s možností jejich úpravy, mazání či vytvoření nové bezdrátové sítě.

SSID	Enabled	Network	Actions
ssid_1	✓	network_1	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
ssid_2	✓	network_2	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Obr. 5.10: Přehled definovaných bezdrátových sítí v kontroléru.

Stránka 5.11 s editací nebo definováním nové sítě obsahuje formulář, pomocí něhož uživatel definuje parametry bezdrátové sítě:

- **SSID** – název bezdrátové sítě tak, jak se zobrazí klientům.
- **Enabled** – přepínač indikující zda bude tato bezdrátová síť povolena.

- **Security** – typ zabezpečení bezdrátové sítě. Pokud je vybráno zabezpečení **WPA Personal**, musí uživatel vyplnit heslo k přístupu k této bezdrátové síti.
- **Network** – dropdown, ze kterého uživatel musí vybrat nadefinovanou lokální síť z 5.3.4, do které bude bezdrátová síť patřit.
- **Hide SSID** – přepínač indikující, zda budou SSID viditelné.
- **Isolate Clients** – přepínač indikující, zda mezi sebou budou moci bezdrátová klienti komunikovat.

The image shows a web form for configuring a wireless network. It consists of several rows:

- SSID**: A text input field containing the placeholder text "wifi network name".
- Enabled**: A checked checkbox.
- Security**: Two radio button options: "Open" (unselected) and "WPA Personal" (selected).
- Password**: An empty text input field.
- Network**: A dropdown menu with a downward arrow and a placeholder "---
- Hide SSID**: An unchecked checkbox.
- Isolate Clients**: An unchecked checkbox.

Obr. 5.11: Formulář pro definování nové bezdrátové sítě v kontroléru.

5.3.6 Seznam asociovaných klientů

Stránka obsahující seznam asociovaných klientů zobrazena na obrázku 5.12 je dostupná pod adresou `/clients`. Obsahuje seznam všech zařízení připojených k bezdrátovým sítím na přístupových bodech OpenWrt, které jsou ve stejné síti jako je kontrolér. Informace o asociovaných zařízeních zahrnují:

- **SSID** – odkaz s názvem bezdrátové sítě, ke které je klient připojen. Po kliknutí na odkaz je uživatel přesměrován na stránku definice této bezdrátové sítě.
- **Client IP Address** – odkaz s IP adresou klienta.
- **Client MAC Adress** – MAC adresa zařízení klienta.
- **Signal/Noise** – síla signálu a šumu na přístupovém bodě OpenWrt, ke kterému je daný klient připojen.

- **TX/RX** – počet přijatých/odeslaných paketů mezi zařízením a přístupovým bodem a aktuální propustnost na přístupovém bodě.

SSID	Client IP Address	Client MAC Address	Signal / Noise	TX	RX
ssid_2	192.168.1.147	AC:AF:B9:5F:4A:5E	-34 / -95 dBm	1.0 MBit/s, 4 Packets	50.8 MBit/s, 17 Packets

Obr. 5.12: Tabulka se seznamem všech asociovaných klientů.

6 Implementace

Tato kapitola se věnuje implementaci práce. Nejprve je popsán datový model využitý na kontroléru, dále jsou popsány typy autentizace ke kontroléru a k zařízením OpenWrt. Následuje popis principu zjišťování dostupnosti zařízení OpenWrt ve sledovaném síťovém segmentu. Nakonec je popsána implementace získávání informace ze sledovaných zařízení a způsob nasazení konfigurace.

Kontrolér byl vyvíjen a testován na zařízení TP-Link TL-WR1043N/ND v4 [21].

6.1 Datový model

Stav a aktuální konfigurace kontroléru je uložena v perzistentní databázi. Pro přístup k databázi a mapování relační databáze na objekty je využita knihovna SQLAlchemy popsaná v kapitole 4.1. Při startu aplikace kontroléru se vytvoří databázový model deklarovaný ve zdrojovém souboru `app/models.py`. Pokud existují uložená data z předchozího spuštění kontroléru, jsou automaticky přidána do databáze. UML diagram databáze kontroléru je zobrazen na obrázku 6.1.

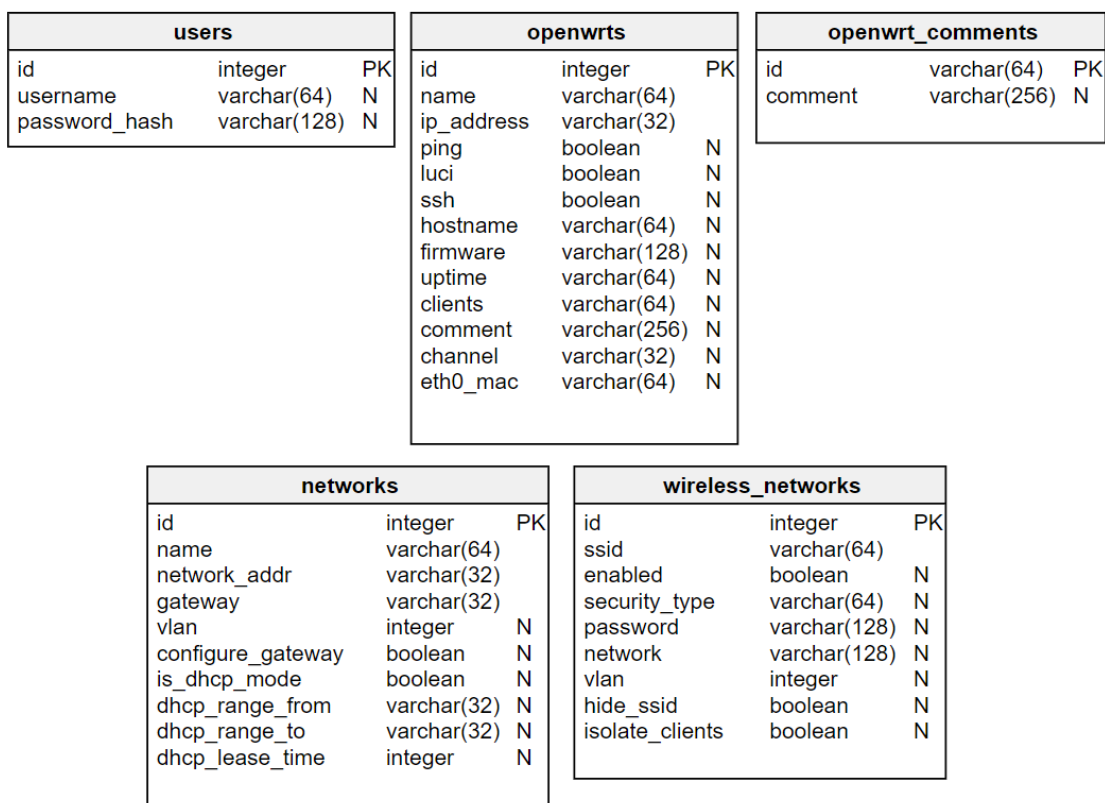
Kontrolér pracuje s následujícími databázovými tabulkami:

- **users** – tabulka obsahující uživatelské přihlašovací údaje ke kontroléru.
- **openwrts** – tabulka obsahující záznamy aktuálních dostupných zařízení OpenWrt ve sledované síti.
- **openwrt_comments** – tabulka obsahující komentáře, které se vážou na konkrétní OpenWrt zařízení přes jejich MAC adresu rozhraní eth0.
- **networks** – tabulka obsahující definované lokální sítě.
- **wireless_networks** – tabulka obsahující definované bezdrátové sítě a jejich parametry.

6.2 Přihlášení uživatele do kontroléru

V současné době se při vývoji webových aplikací klade velký důraz na bezpečnost. V případě kontroléru je důležitá autentizace - ověření identity uživatele.

Identita uživatele je ověřena zadáním platného uživatelského jména a hesla v přihlašovací formě 5.4a. Zaslání vyplněného uživatelského jména a hesla zavolá HTTP metodu POST s daty ve formuláři na URL `'/login'`. Kontrolér se poté pokusí získat záznam o uživateli z databázové tabulky `users` 6.1 a porovná uložené heslo s přijatým heslem. Poté přihlásí uživatele do kontroléru voláním metody `login_user` z knihovny Flask-Login [18], čímž uživateli zašle autentizační data, která se uloží



Obr. 6.1: Diagram databázového modelu kontroléru.

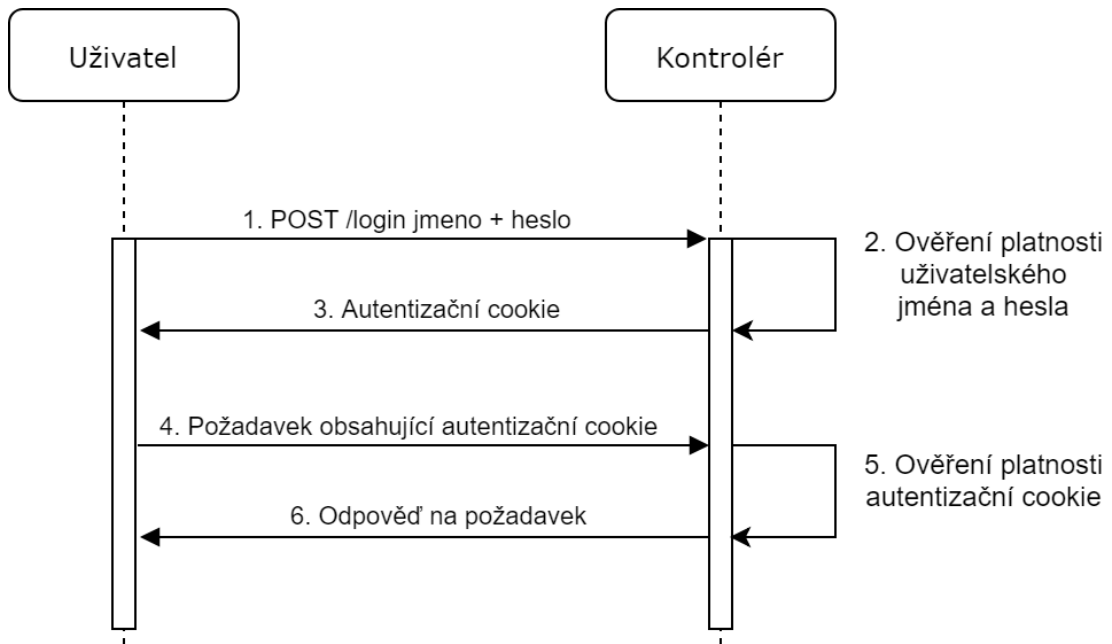
do webového prohlížeče jako cookie ¹, které je použito pro autentizaci všech dalších požadavků uživatele. URL kontroléru, které jsou přístupné pouze přihlášeným uživatelům, obsahují dekorátor `@login_required`. Tento dekorátor znepřístupní dekorovaná URL uživatelům, kteří nejsou přihlášení. Sekvenční diagram znázorňující autentizaci ke kontroléru je uveden na obrázku 6.2.

Uživatelské jméno a heslo nutné pro přihlášení do kontroléru je načteno při startu aplikace přes proměnné prostředí `LOGIN_USERNAME` a `LOGIN_PASSWORD`. Tyto proměnné lze také nastavit v konfiguračním souboru `config.env`.

6.3 Autentizace k zařízení OpenWrt

Většina JSON-RPC volání do zařízení OpenWrt vyžaduje autentizaci klienta, která se ověřuje přidáním autentizačního tokenu do požadavku URL. Autentizační token se získá JSON-RPC voláním knihovny `auth` s požadavkem obsahující následující tělo:

¹Jako cookie se v protokolu HTTP označuje malé množství dat, která WWW server pošle prohlížeči, který si je následně uloží.



Obr. 6.2: Sekvenční diagram autentizace uživatele ke kontroléru.

```
{"method": "login", "params": [<UZIVATEL>, "<HESLO>"]}
```

Autentizační pár <UZIVATEL> a <HESLO> je stejný, jako uživatelské jméno a heslo nutné pro přihlášení do webového rozhraní LuCI na zařízení OpenWrt. V kontroléru jsou přihlašovací údaje nastaveny přes proměnné prostředí `OPENWRT_USERNAME` a `OPENWRT_PASSWORD`. Tyto údaje je také možné definovat v konfiguračním souboru `config.env`.

Po zaslání autentizačního požadavku se správnými přihlašovacími údaji vrátí OpenWrt následující odpověď:

```
{
  "result": "ebdd919a48e721c23ab29748cb32147b",
  "error": null
}
```

Klíč `result` v sobě obsahuje autentizační token, který je nutné přidat jako URL parametr `auth` pro volání všech JSON-RPC metod, které vyžadují autentizaci.

Ve zdrojovém souboru `apps/openwrt_api.py` je definována metoda `call_luci`, která je využívána pro všechna JSON-RPC volání v kontroléru. Zapouzdřuje získání autentizačního tokenu a jeho přidání do URL požadavku.

6.4 Skenování sítě

Skenování sítě se provádí za účelem nalezení všech dostupných OpenWrt zařízení, které může kontrolér spravovat. Síťový segment, na kterém skenování probíhá, je definován systémovou proměnnou `OPENWRT_NETWORK` nebo může být definovaný v konfiguračním souboru `config.env`. Skenují se všechny adresy v daném síťovém segmentu kromě adresy sítě a broadcastové adresy.

Způsob skenování sítě spočívá ve vytvoření vlákna pro každou skenovanou IP adresu v segmentu, kde každé vlákno volá metodu, která se pokusí asynchronně připojit k portu 80 cílové IP adresy. Výpis 6.1 obsahuje kód implementující zkoušku připojení na port 80. Pokud se do sekundy nepodaří připojit k portu 80 na cílové IP adrese, tak se předpokládá, že na dané adrese není dostupné OpenWrt zařízení. Port 80 byl zvolen, jelikož zařízení OpenWrt na tomto portu vždy poslouchá, pokud je aktivní webové rozhraní LuCI a rozhraní JSON-RPC.

Způsob detekce pomocí zkoušky připojení k portu 80 se projevil více spolehlivým a rychlejším, než voláním programu ping. Sken segmentu sítě o rozsahu /24 je vždy do sekundy hotový.

```
1 s~= socket.socket()
2 s.settimeout(1.0)
3 connected = False
4 try:
5     s.connect((ip_address, 80))
6     connected = True
7 except Exception as e:
8     connected = False
9 finally:
10    s.close()
```

Výpis 6.1: Kód zkoušející dostupnost portu 80.

6.5 Získávání informací o OpenWrt zařízeních

Kontrolér získává informace o jednotlivých OpenWrt zařízeních pro sestavení přehledu stavu všech dostupných zařízení, zobrazení detailních informací o konkrétním zařízení a vytvoření seznamu všech asociovaných klientů. Některé informace jdou získat napřímo voláním specifické metody v JSON-RPC rozhraní, jiné informace se musejí získat voláním obecné JSON-RPC metody `exec` knihovny `sys`, přes kterou lze spustit příkaz podobným způsobem jako z příkazové řádky. V tabulce 6.1

Získané informace o zařízení	JSON-RPC Knihovna	JSON-RPC Metoda	JSON-RPC Parametry
Firmware a hardware	sys	exec	ubus call system board
Aktuální čas a stav paměti	sys	exec	ubus call system info
Aktuální zatížení	sys	exec	cat /proc/loadavg
Seznam nakonfigurovaných bezdrátových rozhraní	sys	exec	iwinfo
Systémový log	sys	syslog	-
Nakonfigurovaný WiFi kanál	uci	get	wireless radio0 channel

Tab. 6.1: Způsoby získání hlavních informací ze zařízení OpenWrt.

jsou uvedeny způsoby získání hlavních informací o zařízeních OpenWrt přes rozhraní JSON-RPC.

6.6 Nasazení konfigurace

Princip nasazení konfigurace na zařízení OpenWrt spočívá ve vytvoření změn v konfiguračních objektech UCI modelu zařízení a následném aplikování těchto změn. Tento princip využívá transakční způsob vykonání změn popsany v kapitole 3.1.2. Postupně provedené změny se aplikují až po potvrzení transakce. Postup nasazení konfigurace se skládá z následujících kroků:

1. Smazání existující konfigurace lokálních sítí a nahrání aktuální konfigurace lokálních sítí.
2. Smazání existující konfigurace bezdrátových rozhraní a nahrání aktuální konfigurace bezdrátových rozhraní.
3. Aplikování změn - potvrzení transakce.

V sekcích níže jsou jednotlivé kroky popsány blíže.

Konfigurace sítí

Kontrolér nejprve získá aktuální konfiguraci sítí a interního switche na zařízení pomocí JSON-RPC volání knihovny `uci` s následujícím tělem:

```
{"method": "get_all", "params": ["network"]}
```

Výsledkem tohoto volání je JSON objekt obsahující definice všech nakonfigurovaných sítí a definice interního switche. V dalším kroku se provede iterace nad všemi nakonfigurovanými sítěmi ve vráceném JSON objektu a smazání těch, které

jsou považovány jako výchozí nastavení na zařízení. Názvy sítí, které se zachovávají, jsou `wan_dev`, `lan_dev`, `wan6`, `globals`, `loopback`, `lan`, `wan`. U nastavení interního switche se zachovávají konfigurace pro VID 1 (výchozí lokální síť) a VID 2 (internetová síť).

Smazání konfigurační položky sítě se vykoná JSON-RPC voláním knihovny `uci` s následujícím tělem:

```
{"method": "delete", "params": ["network", <NAZEV_SITE>]}
```

Dále se vytvoří konfigurace interního switche. Pro každý záznam v databázi `networks` 6.1 se vytvoří nová sekce nastavení interního switche v OpenWrt. Sekci definuje JSON s následujícím tělem:

```
{"method": "section",  
  "params": ["network", "switch_vlan", "switch_vlan_<VID>",  
            {"device": "switch0",  
             "ports": "0t 4t",  
             "vlan": <VLAN_INDEX>,  
             "vid": <VID>}]}
```

Níže je blíže popsán význam hlavních konfiguračních páru v JSON objektu.

- **device** – při konfiguraci interního switche je název zařízení vždy `switch0`.
- **ports** – seznam portů, které mají být tagovány/netagovány, zápis `"0t 4t"` znamená, že porty 0 a 4 jsou trunk.
- **vlan** – index VLAN nastavení. Většina OpenWrt zařízení má 2 výchozí VLAN nastavení, takže nová nastavení začínají indexem 3.
- **vid** – VLAN identifikátor, který je definovaný v konfiguraci sítě kontroléru pro každou síť.

Vizualizace nastavení interního switche zařízení OpenWrt modelu TP-Link TL-WR1043N/ND v4 pro záznam s konfigurací `"vid": "10"`, `"ports": "0t 4t"` je zobrazen na obrázku 6.3.

Vytvoření nových VLAN nastavení způsobí vytvoření nových síťových rozhraní. Vytvoření konfigurace s `"vid": "10"` automaticky vytvoří nové síťové rozhraní s názvem `eth0.10`.

Poté se vytvoří konfigurace lokálních sítí, které budou poté svázány s vytvořenými SSID. Podobně jako u konfigurace interního switche OpenWrt se pro konfiguraci lokálních sítí vytvoří nová síť pro každý záznam v tabulce `networks`. Sekci sítě definuje JSON-RPC požadavkem knihovny `uci` s následujícím tělem:

VLAN ID	CPU (eth0)	LAN 1	LAN 2	LAN 3	LAN 4	WAN
Port status:	1000baseT full-duplex	1000baseT full-duplex	no link	no link	no link	no link
<input type="text" value="10"/>	<input type="text" value="tagged"/>	<input type="text" value="tagged"/>	<input type="text" value="off"/>	<input type="text" value="off"/>	<input type="text" value="off"/>	<input type="text" value="off"/>

Obr. 6.3: Vizualizace nastavení interního switchu OpenWrt pro konfiguraci "vid": "10", "ports": "0t 4t".

```

{"method": "section",
 "params": ["network", "interface", "<NAZEV_SITE>",
 {"proto": "none",
 "type": "bridge",
 "ifname": "eth0.+<VID>"}]}

```

Níže je blíže popsán význam hlavních konfiguračních párů v JSON objektu.

- **type** – označuje typ sítě. Takto vytvořená síť bude vždy v bridge módu, což umožní WiFi klientům připojit se do této sítě.
- **ifname** – název síťového rozhraní sítě. Síť je vždy svázána s rozhraním, které bylo vytvořeno po přidání konfigurační sekce interního switchu OpenWrt.

Konfigurace bezdrátových rozhraní

Při nasazení konfigurace bezdrátových rozhraní na zařízení OpenWrt získá kontrolér nejprve aktuální konfiguraci pomocí JSON-RPC volání knihovny uci s následujícím tělem:

```

{"method": "get_all", "params": ["wireless"]}

```

Následně se provede iterace nad aktuálně nakonfigurovanými bezdrátovými rozhraními vrácenými v JSON odpovědi a vytvoří se seznam bezdrátových rozhraní, které budou vymazány. Vymazány budou všechna rozhraní, která nezačínají řetězcem `radio`, jelikož rozhraní s názvem `radio0` bývá výchozí. Vymazání bezdrátového rozhraní se provede JSON-RPC požadavkem s následujícím tělem:

```

{"method": "delete", "params": ["wireless", <NAZEV_ROZHRAANI>]}

```

Dále se nahraje konfigurace všech bezdrátových rozhraní, která jsou uložena v datábazové tabulce `wireless_networks`. Vytvoření bezdrátového rozhraní se provede JSON-RPC požadavkem knihovny uci s následujícím tělem:

```

{"method": "section",
 "params": ["network", "wifi-iface", "<SSID>",
 {"device": "radio0",
 "mode": "ap",
 "ssid": "<SSID>",
 "network": "<NAZEV_SITE>"}]}

```

Níže je blíže popsán význam hlavních konfiguračních páru v JSON objektu.

- **mode** – Označuje v jakém módu bude bezdrátové rozhraní fungovat, v případě konfigurace z kontroléru je to vždy "ap" - přístupový bod.
- **ssid** – Název bezdrátové sítě, která se bude zobrazovat na klientských zařízeních pro připojení.
- **network** – Název sítě svázané s konkrétním bezdrátovým rozhraním. Sítě byly nakonfigurovány v předchozím kroce.

Následně se na vytvořená bezdrátová rozhraní aplikují dodatečná nastavení, jako typ zabezpečení, izolování klientů nebo skrytí vysílaných SSID. Dodatečné parametry se aplikují zasláním JSON-RPC požadavku knihovny uci s následujícím tělem:

```

{"method": "set", "params": ["wireless", <SSID>,
 <NAZEV_NASTAVENI>, <HODNOTA_NASTAVENI>]}

```

Tabulka 6.2 poskytuje přehled konfiguračních možností pro konfigurační páry <NAZEV_NASTAVENI> a <HODNOTA_NASTAVENI>.

Účel	<NAZEV_NASTAVENI>	<HODNOTA_NASTAVENI>
Žádné zabezpečení	"encryption"	"none"
Zabezpečení WPA Personal	"encryption"	"psk2"
Přístupové heslo	"key"	řetězec znaků
Skrytí SSID	"hidden"	"1"nebo "0"
Izolování klientů	"isolate"	"1"nebo "0"

Tab. 6.2: Dodatečné konfigurační parametry bezdrátové sítě zařízení OpenWrt.

Aplikování transakce

Vytvořené změny z předchozích sekcí je nutné aplikovat potvrzením transakce. Transakce se potvrdí zasláním následujících JSON-RPC požadavků knihovny uci:

```

{"method": "commit", "params": ["network"]}
{"method": "commit", "params": ["wireless"]}

```

Tímto jsou provedené změny v definici sítí i bezdrátových rozhraní na zařízení OpenWrt aplikovány.

7 Závěr

V rámci práce byly nejprve definovány důležité pojmy z oblasti bezdrátových sítí standardů IEEE 802.11, problematiky centralizované konfigurace bezdrátových sítí a existující příklady jejich řešení. Dále byla popsána softwarová distribuce OpenWrt, její konfigurační model a dostupné konfigurační nástroje. Následovala kapitola popisující technologie použité v této práci. V další kapitole byl popsán návrh architektury řešení a návrh uživatelského rozhraní. V poslední kapitole byla popsána detailní implementaci výsledného řešení.

Cílem práce bylo vytvořit webovou aplikaci, ze které je možné centralizovaně spravovat přístupové body se systémem OpenWrt. Pro komunikaci s jednotlivými přístupovými body bylo využito rozhraní JSON-RPC, které umožňuje kompletní manipulaci s konfiguračním modelem UCI a také poskytuje přístup ke všem informacím o aktuálním stavu zařízení.

Návrh řešení spočíval v identifikaci konkrétních požadavků kladených na webovou aplikaci. Požadavky byly namodelovány pomocí UML diagramu případů použití. Základní funkční celky, které webová aplikace implementuje, jsou vytváření lokálních sítí, bezdrátových sítí, poskytnutí přehledu o aktuálním stavu jednotlivých přístupových bodů a zobrazení seznamu klientů připojených k bezdrátovým přístupovým bodům.

Pro implementaci byl zvolen lehký webový framework Flask s několika jeho moduly. Pro zajištění moderního a responzivního vzhledu aplikace byla použita CSS knihovna Bootstrap 3 s využitím JavaScriptové knihovny jQuery.

Výsledkem práce je webová aplikace, která umožňuje přehlednou správu a konfiguraci bezdrátových přístupových bodů se systémem OpenWrt. Aplikace je platformě nezávislá, lze jí spustit na jakémkoliv zařízení s nainstalovaným interpretem Python verze 3.7 a potřebnými knihovnami.

Literatura

- [1] OpenWrt - About the OpenWrt/LEDE project [online]. 2018 [cit. 2018-18-11]. Dostupné z URL: <<https://openwrt.org/about>>
- [2] UniFi Controller - User Guide [online]. 2018 [cit. 2018-18-11]. Dostupné z URL: <https://dl.ubnt.com/guides/UniFi/UniFi_Controller_V4_UG.pdf>
- [3] OpenWisp documentation [online]. 2018 [cit. 2018-18-11]. Dostupné z URL: <<http://openwisp.io/docs/index.html>>
- [4] OpenWrt - JsonRpcHowTo [online]. 2018 [cit. 2018-23-11]. Dostupné z URL: <<https://github.com/openwrt/luci/wiki/JsonRpcHowTo>>
- [5] LuCI - Technical Reference [online]. 2016 [cit. 2018-18-11]. Dostupné z URL: <<https://openwrt.org/docs/techref/luci>>
- [6] JSON-RPC 1.0 Specification [online]. 2005 [cit. 2018-18-11]. Dostupné z URL: <https://www.jsonrpc.org/specification_v1>
- [7] OpenWrt - OpenWrt Version History [online]. 2018 [cit. 2018-08-11]. Dostupné z URL: <<https://openwrt.org/about/history>>
- [8] OpenWrt - Build System Usage [online]. 2018 [cit. 2018-08-11]. Dostupné z URL: <<https://openwrt.org/docs/guide-developer/build-system/use-buildsystem>>
- [9] OpenWrt - Installing OpenWrt [online]. 2018 [cit. 2018-08-18]. Dostupné z URL: <<https://openwrt.org/docs/guide-user/installation/generic.flashing>>
- [10] OpenWrt - First Login [online]. 2018 [cit. 2018-08-18]. Dostupné z URL: <<https://oldwiki.archive.openwrt.org/doc/howto/firstlogin>>
- [11] OpenWrt - Packages [online]. 2018 [cit. 2018-08-18]. Dostupné z URL: <<https://openwrt.org/packages/start>>
- [12] OpenWrt - UCI (Unified Configuration Interface) – Technical Reference [online]. 2018 [cit. 2018-08-18]. Dostupné z URL: <<https://openwrt.org/docs/techref/uci>>
- [13] O'HARA, Bob a Al PETRICK. *IEEE 802.11 handbook: a designer's companion*. 2nd ed. New York, NY: IEEE, c2005. ISBN 978-0738144498.

- [14] GAST, Matthew. *802.11 wireless networks: the definitive guide* 2nd ed. Farnham: O'Reilly, 2005. ISBN 0596100523.
- [15] GRINBERG, Miguel. *Flask web development*. Sebastopol, CA: O'Reilly, 2014. ISBN 978-1449372620.
- [16] PILGRIM, Mark. *Ponořme se do Python(u) 3: Dive into Python 3*. Praha: CZ.NIC, c2010. CZ.NIC. ISBN 978-80-904248-2-1.
- [17] CHAFFER, Jonathan a Karl SWEDBERG. *Mistrovství v jQuery: [kompletní průvodce vývojáře]*. Brno: Computer Press, 2013. Mistrovství. ISBN 978-80-251-4103-8.
- [18] Flask-Login [online]. 2019 [cit. 2019-04-18]. Dostupné z URL: <<https://flask-login.readthedocs.io/en/latest/>>
- [19] Flask-SQLAlchemy - Documentation (2.x) [online]. 2019 [cit. 2019-04-20]. Dostupné z URL: <<https://flask-sqlalchemy.palletsprojects.com/en/2.x/>>
- [20] OpenWrt - Switch Configuration [online]. 2019 [cit. 2019-05-02]. Dostupné z URL: <https://openwrt.org/docs/guide-user/network/vlan/switch_configuration>
- [21] Techdata: TP-Link TL-WR1043ND v4.x [online]. 2019 [cit. 2019-05-02]. Dostupné z URL: <https://openwrt.org/toh/hwdata/tp-link/tp-link_tl-wr1043nd_v4>

Seznam symbolů, veličin a zkratek

HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
IEEE	Institute of Electrical and Electronics Engineers
JSON-RPC	Javascript Object Notation over Remote Procedure Call
WLAN	Wireless Local Area Network
DSSS	Direct Sequence Spread Spectrum
FHSS	Frequency Hopping Spread Spectrum
OFDM	Orthogonal Frequency Division Multiplexing
MIMO	Multiple-input, multiple-output
DMG	Directional Multi-Gigabit
BSS	Basic Service Set
SSID	Service Set Identifier
ESS	Extended Service Set
VLAN	Virtual Local Area Network
SDN	Software Defined Network
UCI	Unified Configuration Interface
LuCI	Lua Unified Configuration Interface
UML	Unified Modeling Language
URI	Unified Resource Identifier
URL	Unified Resource Locator
WSGI	Web Server Gateway Interface
DOM	Document Object Model
VID	VLAN Identifier
IP	Internet Protocol
WPA	WiFi Protected Access
MAC	Media Access Control

Seznam příloh

A	Instalace kontroléru	56
B	Adaptace nového zařízení OpenWrt	58
C	Obsah přiloženého CD	59

A Instalace kontroléru

Kontrolér je založený na Python frameworku Flask, takže jej lze spustit z jakékoli platformy, na které je nainstalovaný Python.

Požadavky

- Python 3.7, pip3

Instalace

Nejprve je nutné si naklonovat adresář projektu:

```
$ git clone https://github.com/pjasicek/openwrt-acp
$ cd openwrt-acp
```

Poté stáhnout potřebné Python balíčky:

```
$ pip3 install -r requirements.txt
```

Spuštění

Windows

```
$ set FLASK_APP=main.py
$ set FLASK_ENV=development
$ set FLASK_DEBUG=0
$ python -m flask run --with-threads
```

Linux

```
$ FLASK_APP=main.py
$ FLASK_ENV=development
$ FLASK_DEBUG=0
$ flask run --with-threads
```

Konfigurace

Konfigurace kontroléru se provádí v konfiguračním souboru `config.env`. Při změně některých konfiguračních možností je nutné restartovat aplikaci kontroléru. Níže jsou popsány konfigurační možnosti:

```
# Uživatelské jméno a heslo pro přihlášení do kontroléru
LOGIN_USERNAME=admin
LOGIN_PASSWORD=admin

# Uživatelské jméno a heslo pro přihlášení do webového rozhraní LuCI.
# Mělo by to být stejné jako pro SSH
OPENWRT_USERNAME=root
OPENWRT_PASSWORD=root

# SSH klíč k OpenWrt zařízením, využívá se pro kontrolu dostupnosti SSH
OPENWRT_SSH_KEYFILE=data/openwrt_key.priv

# Sledovaný síťový segment
OPENWRT_NETWORK=192.168.1.0/24

# Interval, ve kterém se automaticky skenuje síť
OPENWRT_SCAN_INTERVAL_SECONDS=900
```

B Adaptace nového zařízení OpenWrt

Pokud je OpenWrt zařízení úplně nové (po nainstalování základního firmwaru), je nejprve nutné vytvořit přihlašovací údaje. To se dá nejjednodušejší udělat přes SSH. Výchozí IP adresa pro OpenWrt zařízení bývá většinou 192.168.0.1 nebo 192.168.1.1. Pro první přihlášení není potřeba heslo.

```
ssh root@192.168.1.1
```

Následně zobrazí OpenWrt výzvu pro zadání nového hesla - je nutné zadat stejné heslo, jako je v konfiguračním souboru kontroléru pod názvem `OPENWRT_PASSWORD`. Po vytvoření přihlašovacích údajů je nutné nainstalovat balíček poskytující podporu pro rozhraní JSON-RPC:

```
opkg install luci-mod-rpc  
/etc/init.d/uhttpd restart
```

Nyní je nutné přenastavit výchozí IP adresu na nějakou adresu ze síťového segmentu kontroléru. Je nutné si dát pozor na potencionální konflikt adres s jinými zařízeními v síti. Poté stačí OpenWrt zařízení zapojit do stejné sítě jako je kontrolér a při dalším skenu sítě v kontroléru se zařízení automaticky detekuje.

C Obsah příloženého CD

Složky se zdrojovými kódy aplikace

- adresář sources – webová aplikace kontroléru

Složky se zdrojovými soubory této práce

- adresář tex