Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Inžinierske dielo

Dalibor Turay, Kristián Košťál, Patrik Krajča, Patrik Pernecký, Peter Radványi, Roman Kopšo, Vladimír Čápka

Študijný program: Softvérové inžinierstvo

Ročník: 1, Krúžok: Po 16:00, U120

Predmet: Tímový projekt

Vedúci: Ing. Rastislav Bencel

Ak. rok: 2015/16

Obsah

1 Big picture

1.1 Úvod

Táto dokumentácia bola vytvorená, aby slúžila na popísanie inžinierskeho diela, ktoré je vyvíjané naším tímom *inWifi*. Hlavným cieľom nášho projektu je analyzovať a pochopiť, akým princípom fungujú softvérovo definované siete (SDN), softvérové kontrolóry v týchto sieťach a spôsob komunikácie wifi prístupových bodov tzv. access pointov (AP). Po tejto analýze vytvoríme jednoduchú sieťovú architektúru, ktorá bude fungovať na princípe SDN a do nej implementujeme viaceré funkcionality spomedzi ktorých, hlavnú funkcionalitu predstavuje plynulý prechod koncového wifi zariadenia medzi dvoma AP. Ide nám o to, aby používateľ tejto siete nezaregistroval zmenu toku prúdenia dát medzi koncovým zariadením a meniacimi sa AP. Kvôli tomuto faktu sme sa nazvali *inWifi*, čo vlastne predstavuje skratku *invisible Wifi*, teda neviditeľná Wifi sieť.

V dnešnom svete tento plynulý prechod ešte nie je plne funkčný a jeho nasadenie v SDN sieťach je ešte menej preskúmané, ako nasadenie v štandardných sieťach. Tento projekt pre nás predstavuje určitú výzvu, lebo môžeme povedať, že naša práca má výskumný charakter. Veľa práce budeme musieť venovať naučeniu sa a pochopeniu nových technológií, aby sme sa vedeli dobre pohybovať v tejto doméne. Okrem iného, sa budeme snažiť, aby prechody nastali bez konfigurácie koncového zariadenia. Ak budeme úspešní, budeme môcť poskytovať klientom vysoko kvalitné sieťové služby, čo bude viesť k ich vyššej spokojnosti. Naše riešenie by sa dalo využiť všade tam, kde je potrebná nepretržitá wifi komunikácia. Napríklad skladoví roboti, ktorí by boli ovládaní wifi AP, by nemali problém s výpadkami komunikácie pri pohybe.

Cieľom prvého semestra pre nás bude analyzovať rôzne SDN technológie, komunikačné protokoly wifi zariadení a iné technológie, ktoré použijeme pre splnenie požiadaviek projektu. Vytvoríme testovacie prostredie, respektíve architektúru, na ktorej budeme môcť testovať a pozorovať, akí sme úspešní, čo sa týka plynulosti prechodu.

Cieľom druhého semestra bude hlavne tvoriť softvér pre SDN a robiť veľa testov, ktorých úspešnosť budeme porovnávať. Na konci vyberieme najúspešnejšiu softvérovú implementáciu SDN, ktorá ponúka najrobustnejšie fungovanie a plynulý prechod. Podľa toho ako sa nám bude dariť, máme aj vedľajšie ciele a tie predstavujú pridávanie nových funkcií do našej SDN architektúry. Týkajú sa hlavne bezpečnosti a jednoduchosti ovládania siete.

1.2 Globálne ciele pre zimný semester

- Rozdelenie rolí v rámci projektu.
- Oboznámenie sa s existujúcimi riešeniami roamingu v sieťach.
- Oboznámenie sa s SDN sieťami, a ako tieto siete fungujú.
- Podrobné oboznámenie sa s Wifi technológiou.
- Oboznámenie sa so štandardami Wifi technológie (autentifikácia, bezpečnosť, atď.)
- Vytvorenie testovacieho prostredia pre SDN siete.
- Simulácia prechodu medzi AP (roaming) v SDN sieťach.
- Pripraviť zariadenia (AP) pre vykonávanie roamingu v SDN sieti.
- Čiastočné vytvorenie prototypu pre roaming.

1.3 Rámcové ciele z pohľadu šprintov

1. Šprint: Ethernet – Oboznámenie sa s Wifi roamingom a SDN sieťami.
2. Šprint: IrDA – Spojazdnenie AP a kontrolóra, vytvorenie metodík, definovanie rizík.
3. Šprint: Bluetooth – Návrh architektúry, testovanie integrácie Wifi s kontrolórom.
4. Šprint: Wifi – Vytvorenie prototypu.
5. Šprint: WiMAX – Vytvorenie a testovanie prototype.

2 Šprint 1 – Ethernet

2.1 Analýza softvérových kontrolórov

2.1.1 Úloha

Hlavnou úlohou tohto šprintu bola analýza dostupných open-source softvérových kontrolórov. Softvérový kontrolór je softvérová platforma ktorá nám umožňuje spravovať a kontrolovať sieť. Tento SDN (Software Defined Networking) softvérový kontrolór musí spĺňať základnú požiadavku a to musí podporovať OpenFlow verziu 1.3. Všeobecne platí, že SDN kontrolór je "mozgom" v prostredí SDN, ktorý oznamuje informácie "dole" k prepínačom a smerovačom zo southbound API a "hore" do aplikácií a obchodnej logiky zo northbound API.

2.1.2 Analýza

Analyzovali sme tieto softvérové kontrolóri:

- NOX bol prvým Openflow kontrolórom avšak podporoval iba OpenFlow 1.0 a to znamená, že nespĺňa naše kritéria.
- POX je všeobecný SDN kontrolór, ktorý podporuje OpenFlow. Ma vysokú úroveň SDN API vrátane grafov topológií a podpory pre vizualizáciu. Taktiež nespĺňa kritéria, pretože je príliš novým na trhu a nemá zatiaľ dostatočnú komunitu.
- OpenDaylight je open-source projekt ktorého cieľom je urýchliť prijatie SDN a vytvoriť pevný základ pre sieťové virtualizácie (NFV). Tento softvér sme odmietli z dôvodu zlých skúseností minuloročného bakalára.
- FlowVisor je kontrolór určený na špeciálny účel, kde sa chová ako transparentné proxy medzi OpenFlow switchmi a viacerými OpenFlow kontrolórmi. Nespĺňa požiadavky, pretože slúži iba na špeciálne účely.
- OpenContrail systém je rozšíriteľná platforma pre SDN, avšak ma iba veľmi slabú dokumentáciu, taktiež nespĺňa kritéria.
- The Floodlight Open SDN kontrolór je kontrolór podnikovej triedy, licencovaný Apachom a založený na jave. Je príliš komplikovaný, čo znamená časovo náročný na pochopenie a preto nespĺňa požiadavky. Ale je zaradený ako plán B.
- Beacon kontrolór je prepojený s Floodlight a preto taktiež nevyhovuje.
- Ryu je open-source SDN kontrolór dizajnovaný na zvýšenie agilnosti siete tým, že sa dá jednoducho spravovať a prispôsobiť ako sa zaobchádza s prevádzkou siete. Tento SDN

kontrolór sme si vybrali pre jeho veľkú komunitu a obsiahlu dokumentáciu a spĺňa hlavnú požiadavku a to, že podporuje OpenFlow 1.3.

2.2 Inštalácia virtuálneho serveru a implementovanie webovej stránky

2.2.1 Úloha

Inštalácia virtuálneho serveru a implementácia Webovej stránky.

2.2.2 Implementácia

Nainštalovali sme virtuálny server, ktorý beží na operačnom systéme Linux Ubuntu. Medzi prvými požiadavkami bolo nainštalovať SSH klienta. Pre správnu funkčnosť webovej stránky sme potrebovali serverovú aplikáciu, ktorá bude podporovať PHP, HTML, CSS, Javascript, jQuery. Jednoznačne sme zvolili Apache HTTP server, ktorý sme nainštalovali príkazom:

sudo apt-get install apache2

Po nainštalovaní Apachu, bolo potrebné dorobiť zopár posledných konfiguračných nastavení. Medzi tieto nastavenia patrí: nastavenie domovského priečinku pre indexový súbor index.html/php, nastavenie práv priečinku a dodatočná inštalácia emailového serveru, aby bolo možné odosielať správy z našej webovej stránky. Naša webová stránka bola naprogramovaná pomocou najpopulárnejšieho HTML/CSS/JS frameworku Bootstrap 3.3.5.

Súčasťou webovej stránky je aj kontaktný formulár implementovaný v PHP. Formulár slúži na kontaktovanie členov tímu priamo z webovej stránky. Členovia správu dostanú v podobe e-mailovej správy. Aby to tak fungovalo, bolo potrebné zabezpečiť aj podporu zo strany operačného systému bežiaceho na serveri. Operačný systém musí byť na to pripravený, aby vedel prijímať požiadavky na odosielanie e-mailových správ od PHP servera a následne pomocou niektorého SMTP servera splniť tieto požiadavky. Ako SMTP klient sme zvolili program *ssmtp*, ktorý sme nainštalovali príkazom:

sudo apt-get install ssmtp

Následne sme upravili konfiguračný súbor klienta, ktorý sa nachádza v adresári */etc* . Po spustení služby *ssmtp* sme funkcionalitu kontaktného formulára na webovej stránke overili posielaním testovacej správy, ktorú úspešne dostal každý člen nášho tímu.

2.3 OpenVSwitch

2.3.1 Úloha

Analýza Open vSwitchu.

2.3.2 Analýza

Open vSwitch je softvérový viacvrstvový prepínač licencovaný pod Open Source Apache 2.0 licencií. Je navrhnutý tak, aby umožňoval masívnu automatizáciu sietí pomocou programových rozšírení. Väčšina zdrojového kódu je napísaná v natívnom jazyku C a je jednoducho prenositeľný do rôznych prostredí, kam patria predovšetkým aj vnorené systémy.

Aktuálna verzia Open vSwitch (v2.4.0) podporuje nasledovné vlastnosti:

- Monitorovanie komunikácie medzi virtuálnych systémov (inter-VM) cez protokolov NetFlow, sFlow(R), IPFIX, SPAN, RSPAN a GRE tunelov.
- LACP (IEEE 802.1AX-2008)
- Štandard 802.1Q – podpora VLAN pomocou trunk liniek
- Multicast snooping
- IETF Auto-Attach SPBM a podpora LLDP
- Štandard 802.1ag pre správu a údržbu sietí
- STP (IEEE 802.1D-1998) a RSTP (IEEE 802.1D-2004)
- Konfigurácia QoS a riadenie premávky
- Podpora pre HFSC qdisc
- Riadenie premávky medzi VM rozhraniami
- NIC bonding with source-MAC load balancing, active backup, and L4 hashing
- Podpora protokolu OpenFlow (s mnohými rozšíreniami pre virtualizáciu)
- Podpora IPv6
- Tunelovacie protokoly (GRE, VXLAN, STT, a Geneve, s IPsec podporou)
- Protokol na vzdialenú konfiguráciu pomocou C a Python väzieb
- Prepínanie (forwarding) v rámci jadra (kernel) a používateľského priestoru (user-space)
- Abstraktná vrstva prepínania (forwarding) umožňuje jednoduchú prenositeľnosť do nových softvérových a hardvérových platforiem

Hlavnou výhodou *Open vSwitch* je to, že podporuje naraz niekoľko verzií protokolu OpenFlow a je kompatbilný s firmvérom *OpenWrt* pre SOHO smerovačov. Súčasná podpora jednotlivých verzií OpenFlow vyzerá nasledovne:

Verzie *Open vSwitch*	Verzie *OpenFlow*					
	1.0	1.1	1.2	1.3	1.4	1.5
1.9 a staršie	áno	nie	nie	nie	nie	nie
1.10	áno	nie	čiastočne	čiastočne	nie	nie
1.11	áno	nie	čiastočne	čiastočne	nie	nie
2.0	áno	čiastočne	čiastočne	čiastočne	nie	nie
2.1	áno	čiastočne	čiastočne	čiastočne	nie	nie
2.2	áno	čiastočne	čiastočne	čiastočne	čiastočne	čiastočne
2.3	áno	áno	áno	áno	čiastočne	čiastočne
2.4	áno	áno	áno	áno	čiastočne	čiastočne

Tab.č.1 - *Stav podpory jednotlivých verzií OpenFlow*

Posledné verzie už majú plnú podporu pre OpenFlow v1.3, ktorá je pre náš projekt postačujúca. Preto vznikol nápad využiť Open vSwitch v prepínačoch SDN v rámci riešenia projektu.

2.4 Rozšírenie MiniNet-u pridaním podpory pre WiFi

2.4.1 Úloha

Úlohou je rozšíriť MiniNet o modul Wifi.

2.4.2 Analýza

Mininet-WiFi je vetva MiniNet-u, ktorá je rozšírená s podporou pre bezdrôtové siete WiFi. Sú pridané virtuálne stanice (STA) a prístupové body (AP) na základe známeho ovládača mac80211/SoftMac. V súčasnosti väčšina bezdrôtových ovládačov na Linuxe používa mac80211/SoftMac, ktorý podporuje podstatnú časť funkcií skutočného bezdrôtového WiFi adaptéra (NIC) a pre Mininet-WiFi umožňuje simuláciu bezdrôtových sietí aj na nižších vrstvách sieťovej architektúry.

Vývoj na Mininet-WiFi sa aj v súčasnosti prebieha ako čistý doplnok emulátora MiniNet pridaním nových abstrakcií a tried s cieľom zabezpečenia podpory virtuálnych bezdrôtových rozhraní a emulovaných liniek pričom funkcionality ako natívna simulácia a podpora pre OpenFlow ostanú nezmenené.

Na projekte aktívne pracujú členovia tímu INTRIG na univerzite Campinas v Brazílie.

2.4.3 Postup inštalácie Mininet-WiFi

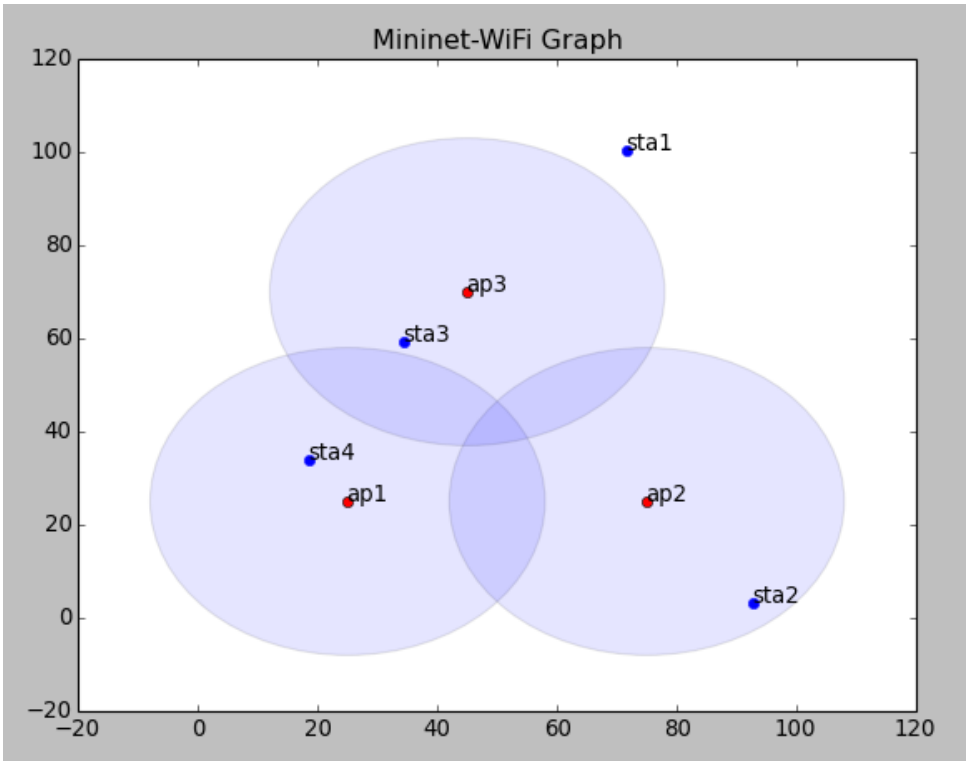
Nasledovný postup bol otestovaný na operačnom systéme Ubuntu v14.04 a na Windowse pomocou virtuálneho stroja, kde bol spustený tiež Ubuntu v14.04. Počas inštalácie sa nainštalujú všetky potrebné balíky a skompiluje sa zdrojový kód stiahnutý z GitHub projektu. Inštaláciu uľahčuje automatizovaný skript, kde výsledkom bude nainštalované prostredie MiniNet rozšírené s podporou pre bezdrôtové siete WiFi.

Na konzole musia byť vykonané nasledovné príkazy za sebou:

```
$ sudo apt-get update
$ sudo apt-get install git
$ git clone https://github.com/intrig-unicamp/mininet-wifi
$ cd mininet-wifi
$ sudo util/install.sh -Wnfv
```

2.4.4 Skúsenosti s Mininet-WiFi

Počas testovania simulátora boli vyskúšané príklady z priečinka examples, a bol navrhnutý a implementovaný aj vlastný skript, kde sú simulované prechody 4 staníc medzi 3 prístupovými bodmi. Počas simulácie bola zapnutá aj vizualizácia simulácie v reálnom čase, ktorú je vidno na obrázku nižšie:



Obr.č.1 - *Simulácia navrhnutej topológie v Mininet-WiFi*

Počas simulácie bolo zistené, že simulátor má niekoľko nedostatkov, ktoré sú:

- Ping medzi niektorými stanicami niekedy bez dôvodu prestane fungovať
- Simulovaný prechod medzi prístupovými bodmi neodzrkadľuje reálnu situáciu
- Sila signálu nie je vypočítaná
- Rôzne neriešiteľné chyby pri spustení simulácie
- Slabá výkonnosť

Tieto nedostatky ešte môžu byť opravené v neskorších verziách simulátora (momentálne sa prebieha aktívny vývoj s týždennými aktualizáciami), ale zatiaľ nie je postačujúce na to, aby sme ho použili na testovanie nášho projektu.

2.5 Analýza bezpečnosti

2.5.1 Úloha

Naštudovať si a analyzovať štandard 802.1X a 802.11i, základný princíp .

2.5.2 Analýza

Štandard 802.1X

IEEE 802.1X je názov protokolu, ktorý umožňuje zabezpečenie prístupu do počítačovej siete. Pokiaľ sa klient (počítač) pripojí k pripojovaciemu bodu, je po ňom pomocou IEEE 802.1X vyžadovaná autentizácia (napr. používateľské meno a heslo). Pripojený bod blokuje všetok dátový tok klienta do tej doby, než je úspešne autentizovaný. Pre riadenie autentizácie je u klienta používaný suplikant , v pripojenom bode je požadovaná dostatočná podpora.

Štandard 802.11i

IEEE 802.11i-2004, alebo 802.11i skrátene, je zmenou pôvodného štandardu IEEE 802.11, realizovaného ako Wi-Fi Protected Access II (WPA2). Návrh štandardu bol ratifikovaný 24. júna 2004. Táto norma stanovuje bezpečnostné mechanizmy pre bezdrôtové siete, nahradzuje doložku pre krátku autentifikáciu a bezpečnosť z pôvodného štandardu. V procese, novela už nepoužíva rozbité Wired Equivalent Privacy (WEP). WEP nahradili tieto štandardy:

- WPA2 (WI-FI Protected access 2) WPA2 implementuje povinné prvky štandardu IEEE 802.11i. Pridáva k TKIP nový algoritmus CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) založený na AES, ktorý je považovaný za úplne bezpečný.

- WPA2 Enterprise (WPA2) Využíva RADIUS (Remote Authentication Dial In User Service) – užívateľská vytáčaná služba pre vzdialenú autentizáciu. Autentizačná metóda 802.1 X/EAP s predvolenou šifrovacou metódou AES (CCMP), taktiež podporuje aj TKIP. Používanou šifrou je AES a voliteľnou je aj RC4.
- WPA2 Personal (WPA2-PSK) Používa zdieľaný kľúč PSK (Pre-shared key), ktorý sa skladá z frázy od 8 do 63 znakov. Táto fráza sa šifruje TKIP alebo AES algoritmom. Používanou autentizačnou metódou je PSK (Pre-shared key) s predvolenou šifrovacou metódou AES (CCMP). Používanou šifrou je AES a voliteľnou je aj RC4.

Ďalej sme sa zaoberali šifrovacími algoritmami:

- TKIP
- AES
- EAP
- LEAP
- PEAP
- PEAPV0/EAP-MSCHAPV2
- PEAPV1/EAP-GTC
- EAP-TLS
- EAP-TLLS/MSCHAPV2
- EAP-SIM

Podrobnejšia analýza sa nachádza v zdieľanom priečinku *InWifi dokumentácia/Analýza* na Google drive. Je rozdelená do dvoch dokumentov s názvami: *Štandard 802.11x* a *Štandard802.11i.*

2.6 Zhodnotenie šprintu

Prvý šprint bol z pohľadu času veľmi náročný, pretože obsahoval veľmi veľa analýzy. V tíme sme si dohodli základné princípy spolupráce a aj rozdelenie úloh. Hlavným cieľom bolo vybrať správny SDN softvérový kontrolór, v ktorom budeme môcť spravovať a testovať našu sieťovú konfiguráciu. Tento cieľ sa nám podarilo splniť. Ako sekundárny cieľ bol OpenVswitch

Ďalej sme sa venovali bezpečnosti WIFI technológie a prenosom aby sme vedeli zaručiť bezpečný prenos dát. Medzi posledné úlohy patrilo vytvorenie webovej stránky, ktorá slúži na prezentáciu nášho tímu, čo sa nám taktiež úspešne podarilo. Za zmienku stojí aj to, že ako tím sme sa prihlásili do TP cup.

3 Šprint 2 – IrDA

3.1 SDN kontrolór RYU

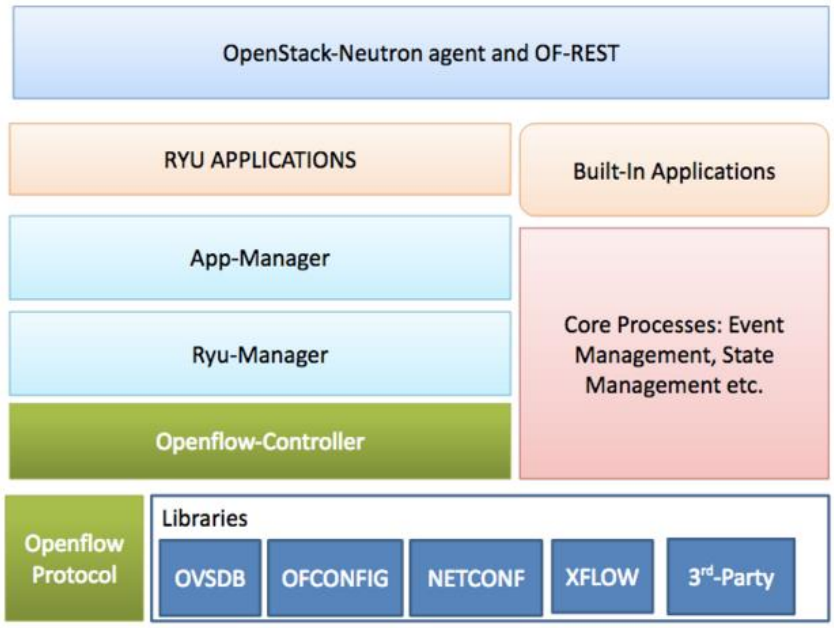
3.1.1 Úloha

Naštudovať si a analyzovať, ako funguje SDN kontrolór RYU.

3.1.2 Analýza

Ryu je open source SDN kontrolór, ktorý ponúka softvérové komponenty používané v aplikáciách, ktoré využívajú technológiu SDN. Vývojári môžu vďaka nemu vytvárať nový manažment siete či riadiace aplikácie. Ryu podporuje rôzne protokoly pre správu siete, napr. OpenFlow protokol (Ryu podporuje aj najnovšiu verziu OpenFlow 1.4). Ryu teda môže vytvárať a posielať OpenFlow správy a taktiež rozoberať a spracovávať prichádzajúce pakety. SDN kontrolór Ryu je postavený na programovacom jazyku Python.

Na obrázku 1 je zobrazená architektúra Ryu.



Obr.č.2 - *Architektúra Ryu* [1]

Ako môžeme vidieť na obrázku, Ryu podporuje veľa rôznych protokolov, okrem OpenFlow aj OF-Config, Open vSwitch Database Management Protocol (OVSDB), NETCONF a XFlow (Netflow and Sflow). Čo sa týka OpenFlow kontrolóra, je to jedna z najdôležitejších súčastí Ryu, pretože je zodpovedný za správu OpenFlow prepínačov. Ryu-Manager slúži na počúvanie na

konkrétnej IP adrese a porte, tak sa môže k nemu pripojiť OpenFlow prepínač. App-Manager je základný komponent pre všetky aplikácie Ryu.

3.2 SDN kontrolór Floodlight

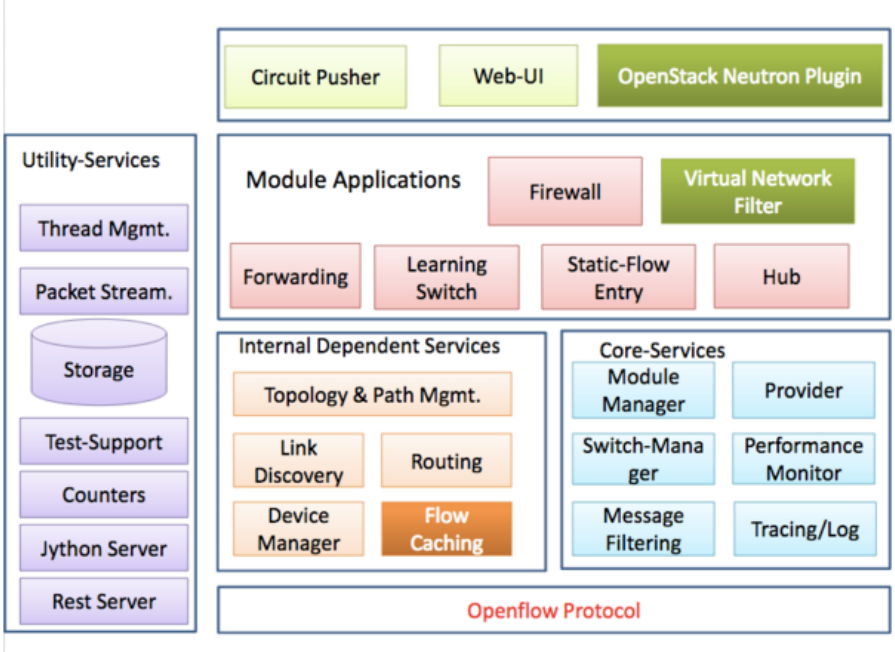
3.2.1 Úloha

Naštudovať si a analyzovať, ako funguje SDN kontrolór Floodlight.

3.2.2 Analýza

Floodlight je open source SDN kontrolór, ktorý je postavený na programovacom jazyku Java. Tento kontrolór tiež podporuje protokol OpenFlow, avšak vie zvládnuť aj zmiešané OpenFlow a non-OpenFlow siete. Floodlight je súčasťou Big Switch projektov a vie pracovať s virtuálnymi aj fyzickými OpenFlow prepínačmi. Floodlight používa ako základ kontrolór Beacon, avšak v porovnaní s týmto kontrolórom sa Floodlight výrazne rozrástol, pričom má dokonalejšie funkcie a lepší výkon.

Floodlight má modulárnu architektúru, pričom zahŕňa rôzne moduly, ako napr. správa topológie, správa zariadení a koncovej stanice, výpočet cesty, infraštruktúru pre prístup na web a iné. Základné komponenty architektúry sú okrem OpenFlow protokolu aplikácie (Rest-API, Module applications) a služby (Core-services, Utility-services, Internal services). Architektúra je zobrazená na obrázku 2.



Obr.č.3 - *Architektúra Floodlight* [2]

3.3 Štandard 802.11k – Assisted roaming

3.3.1 Úloha

Naštudovať si a analyzovať štandard 802.11k, základný princíp a fungovanie tohto štandardu.

3.3.2 Analýza

802.11k redukuje čas roamingu povolením klientovi rýchlejšie určiť, ku ktorému AP sa pripojiť. Hlavný cieľ je dodať inteligentný a optimalizovaný zoznam susedov (Neighbor list) 802.11k podporovaným klientom na optimalizáciu vyhľadávania kanálov, roamingu či využitia batérie. 802.11k povoľuje klientom požiadať o správu obsahujúcu informácie o známych susedných AP, ktoré sú kandidátmi pre roaming.

Klient posiela požiadavku o zoznam susedných AP – tzv. *action paket*, AP odpovie na rovnakej WLAN a rovnakom kanáli – tiež *action paket*. Z tohto paketu potom klient vie, ktoré AP sú kandidátmi pre ďalší roaming. Použitie 802.11k Radio resource management (RRM) umožňuje účinný a rýchly roaming.

Klient teda nepotrebuje všetky z 2,4 alebo 5 GHz kanálov na nájdenie AP – znižuje to využitie kanálov, čím sa zvyšuje bandwith kanálov, redukuje to aj čas a zlepšuje klientove rozhodnutia, zvyšuje životnosť batérie.

Zoznam susedov obsahuje len susedov v rovnakom pásme, obsahuje BSSID, kanál a operačné detaily susedných AP. Použitie zoznamu susedov môže obmedziť potrebu použitia aktívneho či pasívneho skenovania. Zoznam susedov je generovaný dynamicky na požiadanie a nie je udržiavaný na prepínači. Dvaja klienti na rovnakom prepínači, ale odlišných AP môžu mať odlišné zoznamy susedov. Klient posiela požiadavku pre zoznam susedov iba po asociovaní s AP.

Keď prepínač príjme požiadavku o zoznam susedov, prehľadá RRM tabuľku susedov pre zoznam susedov na rovnakom pásme ako AP, s ktorým je klient asociovaný. Potom skontroluje susedov, aktuálne umiestnenie AP, históriu roamingu na prepínači na redukciu zoznamu susedov k 6 na každom pásme.

3.4 Štandard 802.11r – Fast transition roaming

3.4.1 Úloha

Naštudovať si a analyzovať štandard 802.11r, základný princíp a fungovanie tohto štandardu.

3.4.2 Analýza

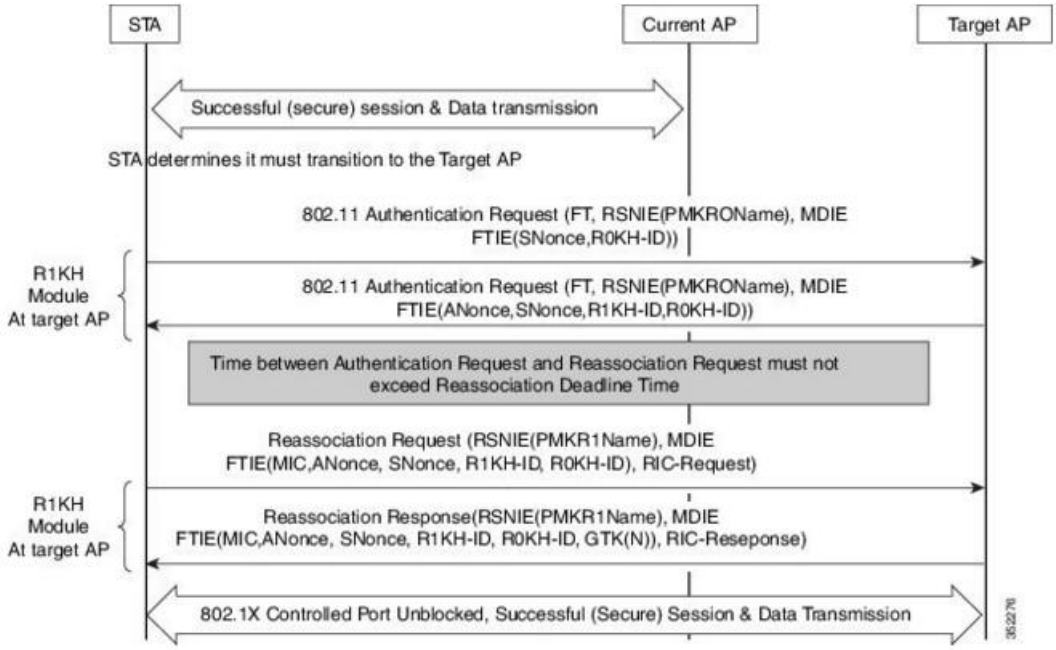
802.11r je štandard IEEE pre rýchly roaming. Tento štandard bol navrhnutý pre bezdrôtové LAN systémy na zrýchlenie autentifikačného procesu.

Handshake s novým AP sa vykoná pred samotným roamingom (fast transition), teda mobilná stanica poskytne novému AP nevyhnutné informácie potrebné na prechod do tohto AP. Handshake dovoľuje AP a klientovi vytvoriť tzv. *Pairwise Transient Key* (PTK) vopred. PTK obsahujú potrebné informácie a sú použité u klienta a AP potom, čo klient robí *re-asociation request*/*response* s novým cieľovým AP. FT key hierarchia dovoľuje klientom robiť BSS (Basic service set) prechody medzi AP bez požadovania autentifikácie na každom AP. Autentifikácia teda prebieha len raz. 802.11r redukuje handoff medzi AP pri poskytovaní QoS a bezpečnosti.

Existujú dve metódy pohybu klienta:

a) **Over the air FT roaming**

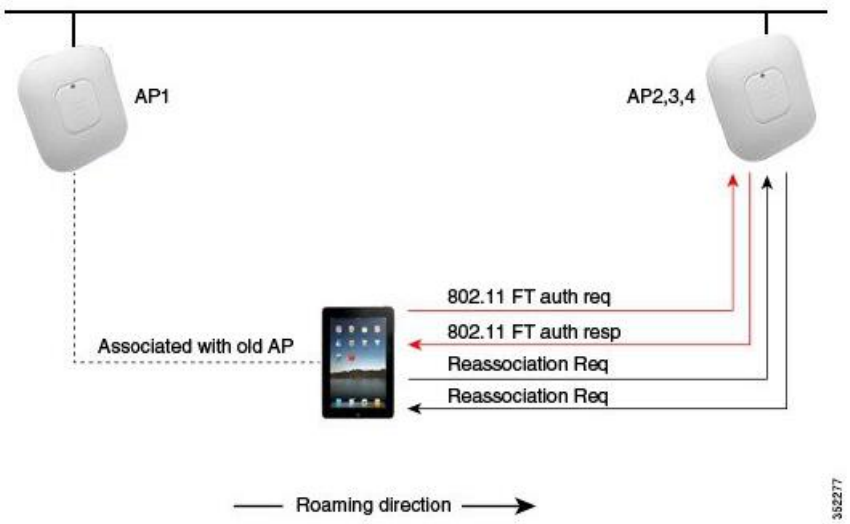
Klient komunikuje priamo s cieľovým AP pomocou IEEE 802.11 autentifikácie použitím FT autentifikačného algoritmu.



Obr.č. 4 - *Over the Air Fast Transition Roaming* [3]

Over the Air Intra Controller Roam – AP1 a AP2 sú pripojené na rovnaký kontrolór.

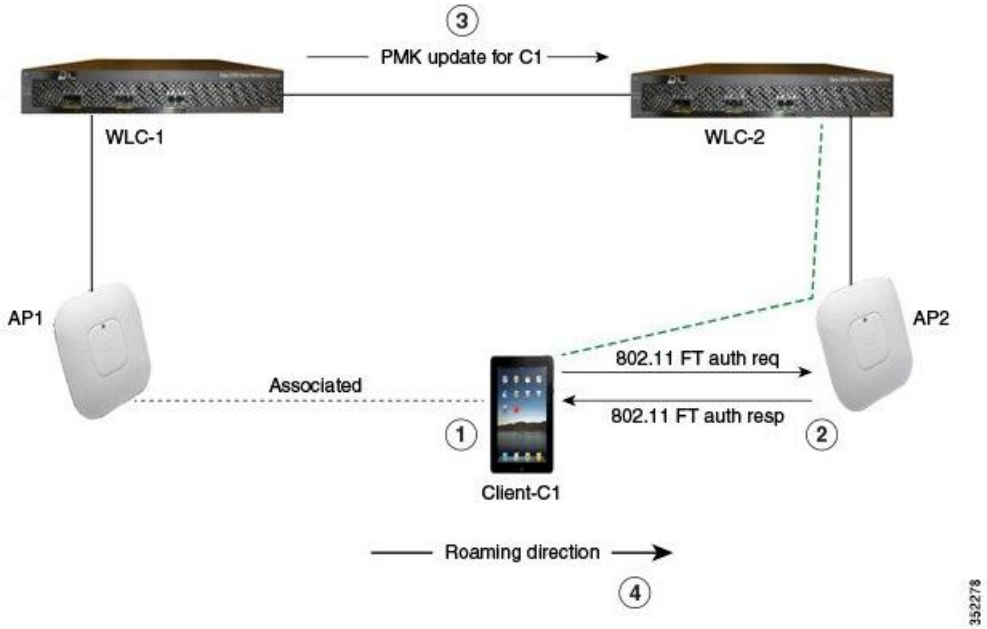
1. Klient je asociovaný s AP1 a chce prejsť do AP2.
2. Klient pošle na AP2 *FT authentication request* a potom od neho príjme *FT authentication response*.
3. Následne pošle na AP2 *reassociation request* a potom od neho príjme *reassociation response*.
4. Klient úspešne prejde do AP2.



Obr.č. 5 - *Over the Air Intra Controller Roam* [3]

Over the Air Inter Controller Roam – AP1 a AP2 pripojené na odlišné kontrolóry.

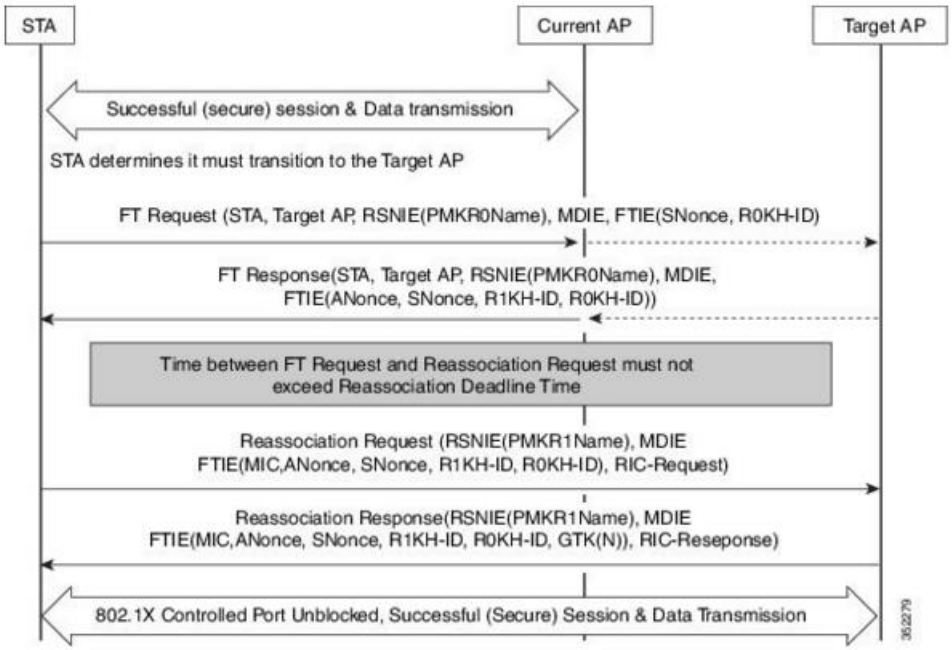
1. Klient je asociovaný s AP1 a chce prejsť do AP2.
2. Klient pošle na AP2 *FT authentication request* a potom od neho príjme *FT authentication response*.
3. Kontrolór, na ktorý je pripojený AP1 pošle *Pairwise Master Key* (PMK) na druhý kontrolór (na ktorom je pripojený AP2).
4. Klient úspešne prejde do AP2.



Obr.č. 6 - *Over the Air Inter Controller Roam* [3]

b) **Over the DS (Distribution System) FT roaming**

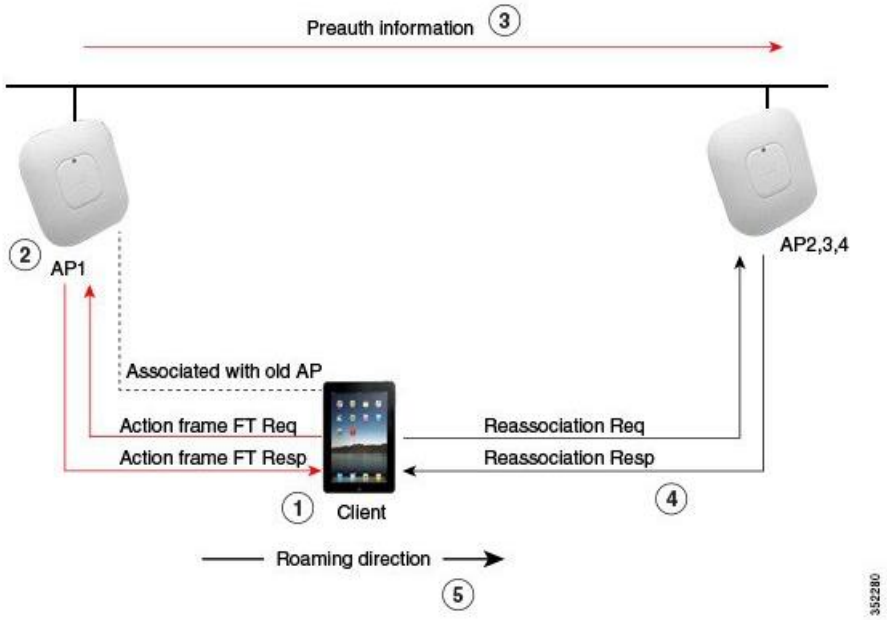
Klient komunikuje s cieľovým AP cez aktuálny AP. Komunikácia sa vykonáva prostredníctvom tzv. *FT action paketov* medzi klientom a aktuálnym AP a potom je poslaná cez kontrolór.



Obr.č. 7 - *Over the DS Fast Transition roaming* [3]

Over the DS Intra Controller Roam – AP1 a AP2 sú pripojené na rovnaký kontrolór.

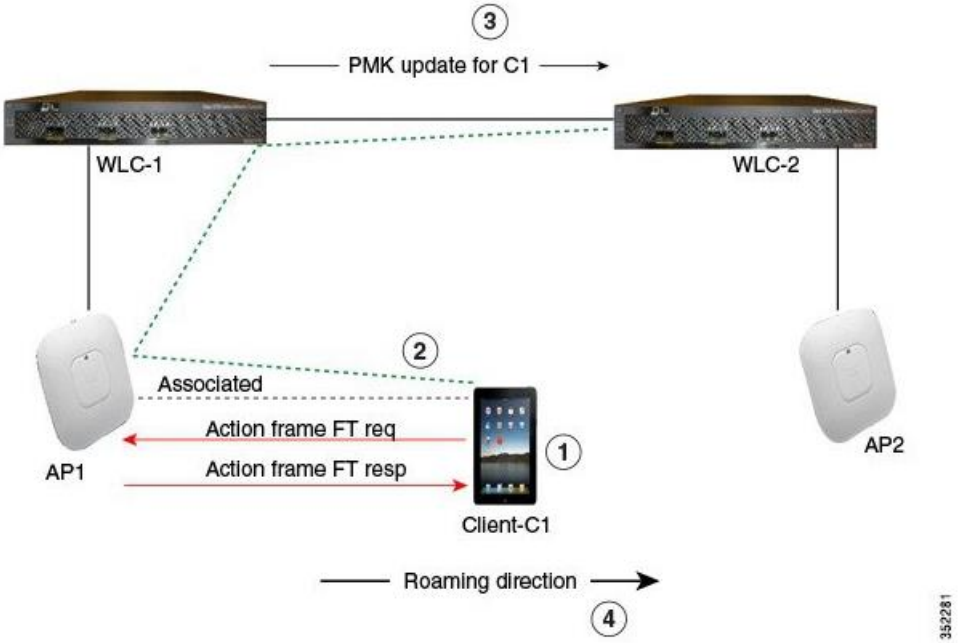
1. Klient je asociovaný s AP1 a chce prejsť do AP2.
2. Klient pošle na AP1 *FT authentication request* a potom od neho príjme *FT authentication response.*
3. Keďže sú oba AP pripojené na rovnaký kontrolór, pred-autentifikačné informácie sú poslané z kontrolóra na AP2.
4. Následne pošle klient na AP2 *reassociation request* a potom od neho príjme *reassociation response*.
5. Klient úspešne prejde do AP2.



Obr.č.8 - *Over the DS Intra Controller Roam* [3]

Over the DS Inter Controller Roam - AP1 a AP2 pripojené na odlišné kontrolóry.

1. Klient je asociovaný s AP1 a chce prejsť do AP2.
2. Klient pošle na AP1 *FT authentication request* a potom od neho príjme *FT authentication response.*
3. Kontrolór, na ktorý je pripojený AP1 pošle *Pairwise Master Key* (PMK) na druhý kontrolór (na ktorom je pripojený AP2).
4. Klient úspešne prejde do AP2.



Obr.č. 9 - *Over the DS Inter Controller Roam* [3]

3.5 Aktualizácia webu

3.5.1 Úloha

Aktualizovať webovú stránku tímového projektu.

3.5.2 Implementácia

Na webovej stránke tímového projektu bol implementovaný mailový server, ďalej bolo zmenené motto nášho tímu, taktiež bol aktualizovaný popis motivácie a cieľu projektu, bol aktualizovaný plán aj dokumenty a nakoniec bola pridaná fotografia nášho tímu.

3.6 Riziká

3.6.1 Úloha

Určiť možné riziká pri práci na projekte, stručne ich popísať, určiť ich pravdepodobnosť, stupeň, dopady a ich mieru. Navrhnúť ošetrenia a vhodné riešenia týchto rizík.

3.6.2 Analýza

Pri práci na projekte boli zistené nasledujúce riziká:

a) **Problém s naštudovaním potrebných materiálov**

Stupeň rizika: 3,6

Stav: Uzatvorené

Popis: Z dôvodu problému k prístupu na server diplomových prác nie je možné študovať efektívne.

Pravdepodobnosť (%): 60

Miera dopadu: 6

Dopady: Študovanie sa výrazne predĺži.

Ošetrenie: Vyžiadanie diplomových prác od vedúceho práce.

Riešenie: Nastal daný scenár a diplomové práce boli úspešne prevzaté od vedúceho práce.

b) **Vybranie nesprávnej aplikácie pre organizáciu**

Stupeň rizika: 1,2

Stav: Uzatvorené

Popis: Existuje mnoho aplikácii na organizáciu projektu agilnou metodou, je možné že si vyberieme nevhodnú alebo aplikáciu ktorá nebude uplne splnať naše požiadavky.

Pravdepodobnosť (%): 30

Miera dopadu: 4

Dopady: Spomalí organizáciu a zapríčiní zmätok v organizácii.

Ošetrenie: Vykonať dôkladnú analýzu týchto webových aplikácii a ich možností.

Riešenie: Vybratie inej webovej aplikácie.

c) **Inštalácia mininetu**

Stupeň rizika: 2,4

Stav: Uzatvorené

Popis: Riziko komplikácii pri inštalácii a nekompatibility mininetu.

Pravdepodobnosť (%): 30

Miera dopadu: 8

Riešenie: Nájdenie inej virtuálnej siete.

d) **Nekompatibilita hardvéru**

Stupeň rizika: 2

Stav: Uzatvorené

Popis: Nekompatibilita smerovača ASUS RT - N16.

Pravdepodobnosť (%): 20

Miera dopadu: 10

e) **Problém so spojazdnením softvérového kontrolóra**

Stupeň rizika: 2,45

Stav: Uzatvorené

Popis: Vybratie nesprávneho softvérového kontrolóra. Existuje mnoho softvérových kontrolórov a nie všetky podporujú OpenFlow 1.3 niektoré sú moc komplikované alebo obsahujú slabú dokumentáciu.

Pravdepodobnosť (%): 35

Miera dopadu: 7

Dopady: Spôsobí predlženie projektu.

Ošetrenie: Vykonať dôkladnú analýzu softvérových kontrolórov a vybratie správneho.

Krízový scenár: Vyberieme iný vhodnejší softvérový kontrolór.

Riešenie:

NOX podporuje iba OpenFlow 1.0 = nespĺňa kritéria

POX je príliš nový a nemá takú komunitu = nespĺňa kritéria

OpenDaylight je náročný na implementáciu = nespĺňa kritéria

Floowvisor sa používa iba na špeciálne účely= nespĺňa kritéria

OpenContrail slabá dokumentácia = nespĺňa kritéria

Floodlight ťažký na naučenie = nespĺňa kritéria

Beacon súvisí s Floodligthom = nespĺňa kritéria

RYU sme vybrali, pretože má veľkú komunitu, rozsiahlu dokumentáciu a podporuje OpenFlow 1.3

f) **Spojazdnenie DD-WRT firmwaru pre router**

Stupeň rizika: 6,3

Stav: Otvorené

Popis: Je potrebné rozchodiť firmware podporujúci OpenFlow 1.3, môžu nastať problémy pri inštalácii alebo kompatibilite.

Pravdepodobnosť (%): 90

Miera dopadu: 7

Dopady: Strata času hľadaním ďalšieho firmwaru.

Ošetrenie: Vybratie iného firmwaru, ktorý bude fungovať správne.

Krízový scenár: V prípade ak nebude fungovať DD-WRT a budú s ním problémy skúsime OpenWRT podporujúci OpenFlow1.3 a ak nebude fungovať správne ani ten, tak použijeme OpenWRT s OpenFlow 1.0.

Riešenie: Vybrali sme OpenWRT s OpenFlow 1.3.

g) **Problém pri spojazdnení virtuálneho stroja**

Stupeň rizika: 2

Stav: Uzatvorené

Popis: Môžu nastať komplikácie v nastavení alebo inštalácii Linuxu, Apache serveru,SMTP serveru.

Pravdepodobnosť (%): 50

Miera dopadu: 4

Dopady: Strata času hľadaním riešenia vzniknutých chýb pri inštalácii alebo pri hľadaní riešenia potrebných nastavení.

Ošetrenie: Hľadanie riešení na fórach alebo v dokumentácii Linuxu, Apache, SMTP.

3.7 Hipchat

Úlohou bolo založiť skupinu na webovej službe HipChat pre udržiavanie a ukladanie komunikácie v tíme.

3.7.1 Analýza

HipChat je webová služba určená na súkromnú komunikáciu (chat). Takisto poskytuje aj úložisko súborov, videohovory, možnosť prehľadávať históriu správ a prezeranie obrázkov. Pre tímový projekt je teda táto služba vhodná, pretože môžeme medzi komunikovať v ľubovoľnom čase, keď

máme prístup k internetu, pričom každá komunikácia je uložená do histórie. Taktiež si môžeme ukladať rôzne súbory, či už zdrojové kódy, dokumentácie a pod.

3.7.2 Implementácia

Pre potreby komunikácie bolo potrebné vytvoriť nejaký profesionálny chatovací prostriedok, ktorý oddelí súkromný život od pracovného, pretože v prípade Facebooku je toto jeho najväčšou nevýhodou. Pri rozhodovaní medzi Hipchat a Slack, čo sú dvaja najväčší konkurenti padlo naše rozhodnutie na Hipchat, veď predsa budeme mať možnosť vyskúšať si Slack napríklad v ďalšom semestri. Pre vytvorenie chatu v Hipchate sú potrebné nasledovné kroky:

- registrácia
- vytvoriť group (rozhodnúť, či bude verejná, verejne viditeľná alebo súkromná)
- pozvať ľudí, spolupracovníkov
- vytvoriť miestnosti na chat
- pridať ľudí
- chatovať, komunikovať, video/audio hovor

3.8 Simulácia v OpenNet

3.8.1 Úloha

Simulácia v simulátore OpenNet.

3.8.2 Implementácia

Ako možné riešenie pre nedostatkov, ktorých sme zistili pri otestovaní simulátora Mininet-WiFi bolo vyskúšanie simulátora OpenNet. Simulátor OpenNet vznikol spájaním emulátora MiniNet so simulátorom NS3 a tým pádom bola dosiahnutá podobná funkcionalita ako pri Mininet-WiFi.

Nový simulátor umožňuje spoľahlivejšiu simuláciu prechodov, avšak nám to ešte nestačí na overenie riešenia s rýchlim prechodom v rámci roamingu. Ďalej neumožňuje priamu doimplementáciu funkcionality centralizovaného riadenia procesov v bezdrôtových prístupových bodoch (AP), čo neumožňuje ani Mininet-WiFi.

Na rozdiel od Mininet-WiFi tento simulátor priamo nepodporuje ani vizualizáciu (animáciu) v reálnom čase.

Na základe týchto nedostatkov pri simulátoroch sme rozhodli naše riešenie testovať už priamo v rámci reálnych zariadení a sieťovej topológie.

3.9 Nasadenie a spojazdnenie SDN kontrolóra a prepínača

3.9.1 Úloha

Spojazdnenie SDN kontrolóra RYU na smerovači.

3.9.2 Implementácia

Ako kontrolór sme sa rozhodli použiť Ryu v najnovšej verzii, ktorá podporuje OpenFlow od 1.0 až do 1.5. Je to open-source kontrolór, do ktorého si vieme doimplementovať potrebné veci, ak by to bolo žiadané. Je implementovaný v jazyku Python a v dnešnej dobe čelí obrovskej popularite z pohľadu SDN. Pre nainštalovanie Ryu potrebujeme počítač s nainštalovanou distribúciou Linux. Podľa nastavení nášho prepínača sa má kontrolór nachádzať na IP adrese 192.168.1.10:6633, takže musíme zmeniť IP počítača tak, aby sedela. Potom cez konzolu spustíme ako super user príkazy:

```
% git clone git://github.com/osrg/ryu.git
% sudo apt-get install python3.5
% cd ryu; python ./setup.py install
```

Potom následne môžeme ryu spustiť a začať komunikáciu. Spustíme cvičný OpenFlow 1.3 skript.

```
% cd bin
% ./ryu-manager ryu/app/simple_switch13.py
```

Teraz máme spustený kontrolór a komunikácia môže začať. Po zapnutí prepínača vidíme, že spolu s kontrolórom komunikujú, dokonca keď zapojíme do smerovača internet, tak všetky správy sú šírené.

3.10 Firmware na smerovač s OpenFlow 1.3

3.10.1 Úloha

Implementovať na smerovač firmware s OpenFlow 1.3.

3.10.2 Implementácia

Podrobné návody sú opísané v dokumente Riadenie projektu v kapitolách Manažment softvéru.

3.11 Metodiky

Všetky metodiky sú popísané v dokumentácii pre riadenie projektu.

3.12 Zhodnotenie šprintu

Druhý šprint bol sčasti analytický, keďže sa museli zanalyzovať SDN kontrolóry RYU a Floodlight, ako aj štandardy 802.11k a 802.11r. Ďalej sa v druhom šprinte spísali potrebné metodiky, aktualizoval sa web, pokračovalo sa v práci na rizikách a taktiež sa vytvoril skupinový chat vo webovej službe HipChat. Z praktických vecí sa v druhom šprinte simulovalo v OpenNet, pričom sa

zistilo, že OpenNet je nevyhovujúci pre naše riešenie. Ďalej sa nasadil a spojazdnil SDN kontrolór na smerovači a taktiež sa naň implementoval aj firmware s OpenFlow 1.3.

4 Šprint 3 – Bluetooth

4.1 Základné bloky architektúry

4.1.1 Úloha

Navrhnúť základných blokov architektúry nášho riešenia. Pri návrhu je potrebné brať do úvahy aj bezpečnosť.

4.1.2 Návrh

Návrh sa nachádza v kapitole *6 Architektúra.*

4.2 Návrh Virtual AP

Na tejto úlohe sa ešte pracuje v rámci 3.šprintu. V čase odovzdania, táto úloha ešte nebola splnená.

4.3 Metodika testovania

4.3.1 Úloha

Navrhnúť správnu metodiku pre otestovanie najmä už pripraveného SDN prepínača a komunikáciu kontrolóra RYU s prepínačom. Vymyslieť si scenáre, ktoré dokážu pokryť všetky možné zdroje chýb.

4.3.2 Návrh

Návrh sa nachádza v dokumente riadenie projektu, v časti Manažment testovania.

4.4 OpenFlow 1.0

4.4.1 Úloha

Pripraviť si topológiu kompatibilnej s protokolom OpenFlow 1.0 podľa definovaných scenárov testovania. Následne prepojiť prepínače so serverom, kde beží softvérový SDN controller RYU. Hardvérový prepínač musí preposielať pakety presne tak, ako je to definované pomocou Python skriptov na controlleri. Ako referenciu je možné použiť skript *simple_switch.py* .

4.4.2 Implementácia

Priamo na našom hardvérovom prepínači bol testovaný firmvér OpenWrt 15.05 Chaos Calmer s implementovaným OpenFlow 1.0 od univerzity v americkom Stanforde. Podrobná implementácia je opísaná v dokumente Riadenie projektu v časti Manažment softvéru. Po testoch bolo zistené, že OpenFlow 1.0 nie je dostačujúci pre náš projekt, pretože nepodporuje viacero flow tabuliek.

4.5 OpenFlow 1.3

4.5.1 Úloha

Pripraviť si topológiu kompatibilnej s protokolom OpenFlow 1.3 podľa definovaných scenárov testovania. Následne prepojiť prepínače so serverom, kde beží softvérový SDN controller RYU. Hardvérový prepínač musí preposielať pakety presne tak, ako je to definované pomocou Python skriptov na controlleri. Ako referenciu je možné použiť skript *simple_switch_13.py* .

4.5.2 Implementácia

Momentálne sa na tejto úlohe stále pracuje.

Podrobný návod ako implementovať OpenFlow 1.3 je v dokumente Riadenie projektu v kapitole Manažment softvéru, kde sú rozobraté dva prípady implementácie OpenFlow 1.3. Nakoniec sme sa rozhodli pre implementovanie softvérového Open vSwitch do smerovača, ktorý zmení smerovač na SDN prepínač (forwarder).

4.6 Flow tabuľky

4.6.1 Úloha

Po úspešnom spustení SDN topológie s niektorou verziou OpenFlow spolu s kontrolórom nasleduje úloha na overenie stavu Flow tabuliek na každom prepínači. Stav Flow tabuliek by mal odzrkadľovať toku dát cez sieť.

4.6.2 Implementácia

Flow tabuľky sa dajú zobraziť na prepínačoch s Open vSwitch vydaním nasledovných príkazov, kde predpokladáme že názov rozhrania bridge máme nastavené na *br0*:

```
ovs-ofctl dump-flows br0          # zobrazí OpenFlow flows a hidden flows
ovs-appctl bridge/dump-flows br0  # zobrazí OpenFlow flows a hidden flows
ovs-appctl dpif/dump-flows br0    # zobrazí informácie o bridge a datapath
ovs-dpctl dump-flows [dp]         # zobrazí informácie o Linux kernel a datapath
```

4.7 Vizualizácia topológie

4.7.1 Úloha

Podľa dostupných informácií softvérový kontrolór RYU umožňuje vizualizáciu sieťovej topológie cez webové rozhranie. Naštudujte si potrebné materiály k vizualizácii a otestujte to v rámci funkčnej topológie z predchádzajúcich úloh.

4.7.2 Implementácia

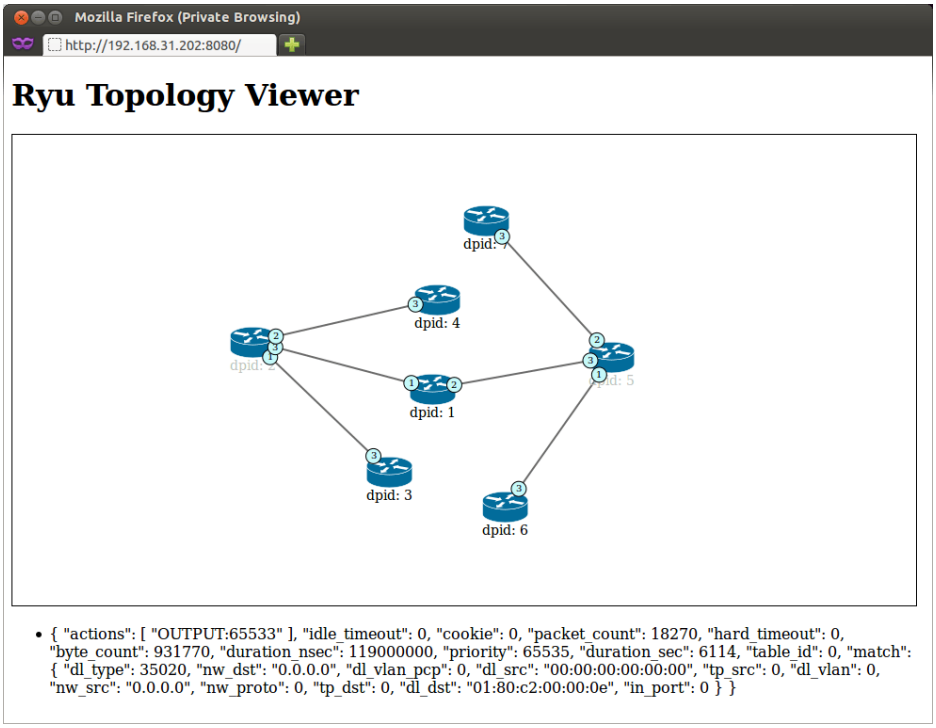
Keď máme našu SDN architektúru korektne zapojenú, môžeme využiť vlažnosti RYU kontrolóra na vizualizáciu našej topológie. Táto vizualizácia závisí od troch aplikácií:

- ryu.app.rest_topology: získa dáta z uzlov a liniek
- ryu.app.ws_topology: notifikuje o zmene spojenia (up/down)
- ryu.app.ofctl_rest: získa dátové cesty tokov

Na terminálovom okne kontrolóra RYU stačí zavolať nasledovný príkaz:

```
$ PYTHONPATH=. ./bin/ryu run --observe-links ryu/app/gui_topology/gui_topology.py
```

Potom na lokálnej adrese RYU host na porte 8080 môžeme cez webové GUI rozhranie vidieť pekne zviditeľnenú našu topológiu ako je vidieť na obrázku 6.



Obr.č.10 – *GUI na vizualizáciu topológie pomocou RYU* [4]

4.8 Zhodnotenie šprintu

Tento šprint bol zatiaľ najnáročnejší, keďže okrem podrobného otestovania dosiaľ hotových častí riešenia sa intenzívne pracovalo aj na dokumentácii projektu. Predovšetkým bola skompletizovaná základná architektúra systému. Následne boli vykonané funkčné testy sieťovej topológie, pripraveného SDN prepínača s OpenWrt a Open vSwitch a komunikácia prepínača s kontrolórom RYU. Boli vyskúšané obidve známe verzie protokolu OpenFlow, ktoré sú 1.0 a 1.3. Počas testovania boli aj sledované zmeny vo Flow tabuľke prepínača. Bolo vyskúšané aj možnosť vizualizácie sieťovej topológie pomocou kontrolóra RYU s určitou úspešnosťou. Podstatnou časťou šprintu bolo aj príprava dokumentácie na odovzdanie, kde sa pracovalo najmä na dokumentácii riadenia a inžinierskom diele.

5 Použité technológie

Aktuálne používané technológie:

a) **Ryu SDN framework**

Oficiálny web: http://osrg.github.io/ryu

Verzia: 3.26

Popis: Sieťový framework definovaný component-based softvérom.

b) **Open vSwitch**

Oficiálny web: http://openvswitch.org/

Verzia: 2.4.0

Popis: Open vSwitch je opensource implementácia distribuovaného virtuálneho viacúrovňového prepínača.

c) **OpenWRT**

Oficiálny web: https://openwrt.org

Verzia: 15.05

Popis: OpenWRT je operačný systém založený na Linoxovom jadre, primárne používaný na zariadeniach na prepínanie sieťovej premávky.

d) **Python**

Oficiálny web: https://www.python.org/

Verzia: 3.5.0

Popis: Python je vysokoúrovňový skriptovací programovací jazyk.

e) **Bootstrap**

Oficiálny web: http://getbootstrap.com/

Verzia: 3.3.5

Popis: Bootstrap je opensource kolekcia pomôcok na vytváranie web stránok a web aplikácií.

f) **Javascript**

Oficiálny web: https://www.javascript.com/

Verzia: ECMA - 262, revízia 5 (V8 engine)

Popis: Javascript je webový programovací jazyk. Väčšinou sa používa vo webových prehliadačoch.

g) **jQuery**

Oficiálny web: https://jquery.com/

Verzia: 2.1.4

Popis: JQuery je javascriptová knižnica s širokou podporou prehliadačov, ktorá kladie dôraz na interakciu medzi JavaScriptom a HTML.

h) **Html5**

Oficiálny web: http://www.w3schools.com/html/default.asp

Verzia: 5.0

Popis: HTML5 je verzia značkovacieho jazyka HTML používaného pre tvorbu webových stránok.

i) **Css3**

Oficiálny web: http://www.w3schools.com/css/default.asp

Verzia: 3.0

Popis: CSS je štandard pre zápis kaskádových štýlov pre web stránky.

j) **PHP**

Oficiálny web: https://secure.php.net/

Verzia: 5.6.15

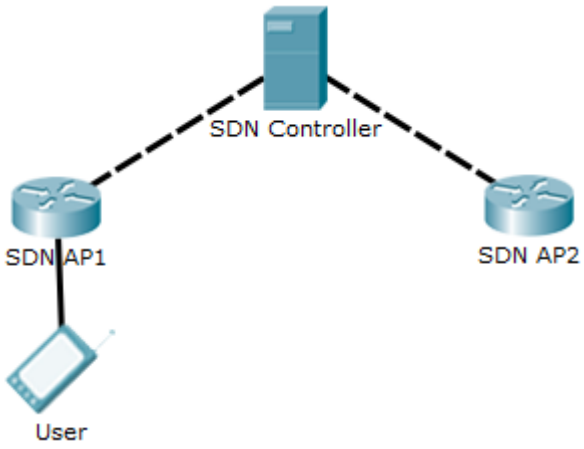
Popis: PHP je populárny opensource skriptovací jazyk, ktorý sa používa najmä na programovanie klient-server aplikácií a pre vývoj dynamických webových stránok.

6 Architektúra

Naším cieľom je vytvoriť zapojenie siete pomocou SDN controllera a access pointov (AP) tak, že pri prechode používateľa medzi dvoma AP nedôjde k strate údajov.

6.1 Návrh základnej architektúry

V základnom návrhu pracujeme bez zabezpečenia siete. Avšak v budúcnosti použijeme na autentifikáciu autentifikačný server, ktorý bude pripojený k SDN controlleru.



Obr.č.11 - *Základný návrh zapojenia siete*

6.2 Návrh jednotlivých častí architektúry

Na obrázku 1 vidíme základný návrh zapojenia siete. SDN controller komunikuje s oboma AP cez ethernetové káble a obidve AP komunikujú s používateľmi pomocou bezdrôtovej siete (wifi).

SDN controller – Slúži na všetko dôležité rozhodovanie v sieti, aby sme mohli použiť lacnejšie AP, ktoré potom nemusia rozhodovať napríklad o smerovaní v sieti. Pri použití SDN controllera nemusia robiť AP zložité rozhodnutia v sieti, a iba preposielajú podľa vopred určených pravidiel ktoré sú uložené vo flow tabuľkách správy. Samotný SDN controller pritom upravuje tieto flow tabuľky, v prípade, že sa chce používateľ pripojiť do siete rozhoduje o tom ku ktorému AP sa má používateľ pripojiť, stará sa o autorizáciu používateľov, a v neposlednom rade sa stará o to kedy má prebehnúť handover používateľa medzi dvoma AP a aj to na ktoré AP sa má potom používateľ pripojiť. Ak v sieti nastanú zmeny, napríklad výpadok AP, všetky rieši controller. Z tohto dôvodu je vhodné, aby bol controller zálohovaný záložným controllerom v prípade, že by hlavný SDN controller nebol funkčný napríklad z dôvodu zlyhania počítača.

Ako SDN controller sme použili RYU controller.

SDN Access Point – Keďže o hlavnej logike siete rozhoduje SDN controller, môžeme použiť lacnejšie AP. O všetkom smerovaní rozhodujú flow tabuľky, ktoré nám vlastne určujú na ktorý port sa má daná správa preposlať, a či má prísť k nejakej úprave, alebo k zahodení správy. V prípade že sa do siete chce pripojiť používateľ posiela beacon správy, ktoré AP preposiela SDN controlleru, ktorý rozhodne či sa môže pripojiť používateľ do siete.
Na AP sme aplikovali po viacerých testoch nakoniec Open vSwitch s open flow 1.3.

User – Používateľ ktorý je pripojený k sieti. Snažíme sa docieliť bezstratový handover bez toho, aby sme museli na strane používateľa niečo meniť, a aby celý tento proces vôbec používateľ nezaregistroval.
Ako používateľa považujeme každé zariadenie, ktoré má možnosť sa bezdrôtovo pripojiť na AP.

7 Literatúra

[1] Shindar, Rao.: SDN Series Part Four: Ryu, a Rich-Featured Open Source SDN Controller Supported by NTT Labs. [online]. THENEWSTACK, 2015. [cit: 2015-16-11]. Dostupné na internete: <http://thenewstack.io/sdn-series-part-iv-ryu-a-rich-featured-open-source-sdn-controller-supported-by-ntt-labs/>.

[2] Shindar, Rao.: SDN Series Part Five: Floodlight, an OpenFlow Controller. [online]. THENEWSTACK, 2015. [cit: 2015-16-11]. Dostupné na internete: <http://thenewstack.io/sdn-series-part-v-floodlight/>.

[3] Cisco.: 802.11r Fast Transition Roaming. [online]. Verzia 3.3. Cisco. [cit: 2015-16-11]. Dostupné na internete: <http://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/5700/software/release/ios_xe_33/11rkw_DeploymentGuide/b_802point11rkw_deployment_guide_cisco_ios_xe_release33/b_802point11rkw_deployment_guide_cisco_ios_xe_release33_chapter_01.html>.

[4] Nippon Telegraph and Telephone Corporation.: Topology Viewer. [online]. Nippon Telegraph and Telephone Corporation, 2011-2014. [cit: 2015-16-11]. Dostupné na internete: <http://ryu-zhdoc.readthedocs.org/en/latest/gui.html>.