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

- 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 1: Oboznámenie sa s Wifi roamingom a SDN sieťami.
2. Šprint 2: Spojazdnenie AP a kontrolóra, vytvorenie metodík, definovanie rizík.
3. Šprint 3: Návrh architektúry, testovanie integrácie Wifi s kontrolórom.
4. Šprint 4: Vytvorenie prototypu.

2 Analýza

2.1 Softvérové kontrolóry

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, že 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 z northbound API.

Analyzovali sme tieto softvérové kontrolóry:

- 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 kritériá, 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 má 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 tomu, 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 OpenVSwitch

Open vSwitch je softvérový viacvrstvový prepínač licencovaný pod Open Source Apache 2.0 licenciou. 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.

OpenWrt je opisovaný ako distribúcia Linux pre vnorené systémy. Namiesto snahy vytvoriť jeden, statický firmvér, OpenWrt poskytuje plne zapisovateľný systém súborov s manažmentom modulov a balíkov. To oslobodzuje od výberu aplikácií a konfigurácií poskytnutých výrobcom a umožňuje upraviť zariadenie prostredníctvom použitia akýchkoľvek balíkov pre danú aplikáciu. Pre vývojára, OpenWrt je vývojové prostredie na vytvorenie aplikácie bez potreby vytvoriť celý nový firmvér; pre používateľa to znamená schopnosť plnej úpravy zariadenia v spôsoboch, o ktorých nikto nevedel.

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

- Monitorovanie komunikácie medzi virtuálnymi systémami (inter-VM) cez protokoly NetFlow, sFlow(R), IPFIX, SPAN, RSPAN a pomocou 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 smerovače. 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.

2.3 Rozšírenie MiniNet-u pridaním podpory pre WiFi

Mininet-WiFi je vetva MiniNet-u, ktorá je rozšírená s podporou pre bezdrôtové siete WiFi. Virtuálne stanice (STA) a prístupové body (AP) sú pridané 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 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.

2.4 Štandard 802.1X

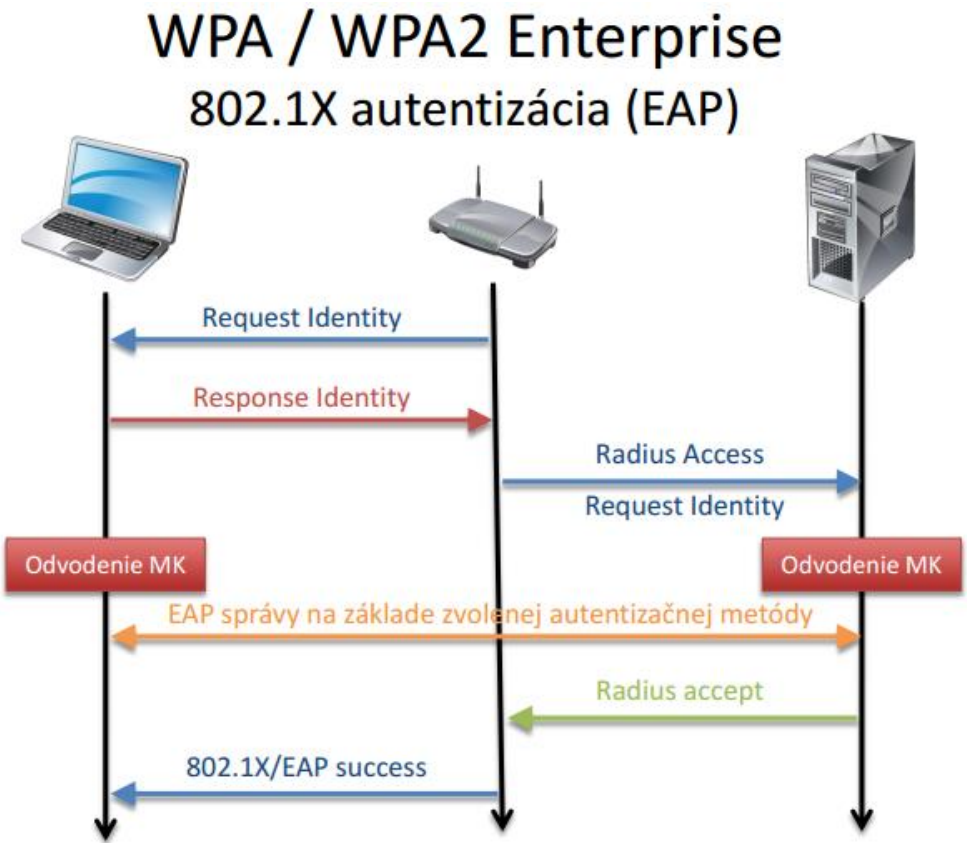
IEEE 802.1X je názov protokolu, ktorý umožňuje zabezpečenie prístupu do počítačovej siete. Pokiaľ sa klient (počítač) pripojí k pripojovaciemu bodu, je po ňom pomocou IEEE 802.1X vyžadovaná autentizácia (napr. používateľské meno a heslo). Pripojený bod blokuje všetok dátový

tok klienta do tej doby, než je úspešne autentizovaný. Pre riadenie autentizácie je u klienta používaný suplikant , v pripojenom bode je požadovaná dostatočná podpora.

Princíp činnosti

Pokiaľ sa používateľ pripojí na sieťový port, má blokovanú všetku komunikáciu okrem EAP protokolu, ktorý zaisťuje autentizáciu. Autentizácia prebieha nasledovne:

1. Klient sa pripojí k prístupovému bodu
2. Prístupový bod akceptuje iba autentizačné EAP rámce
3. ostatný (dátový) tok od klienta je zablokovaný
4. klient odošle autentizačné informácie pomocou EAP protokolu
5. prístupový bod prepošle žiadosť RADIUS serveru
6. na RADIUS serveri prebehne overenie používateľa
    a. pokiaľ je používateľ lokálny prebehne jeho overenie priamo na RADIUS serveri
    b. pokiaľ používateľ nie je lokálny, prebehne žiadosť o autentizáciu cez štruktúru RADIUS serverov až k používateľovej domovskej sieti
7. výsledkom autentizácie je informovaný prístupový bod, ktorý v prípade úspechu odblokuje klientovi dátový tok.



Obr.č.1 - *Autentifikácia pomocou RADIUS servera* [1]

2.5 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).

802.11i nahrádza predchádzajúcu špecifikáciu zabezpečenia, Wired Equivalent Privacy (WEP), pri ktorom bolo preukázané, že má chyby zabezpečenia. Wi-Fi Protected Access (WPA), bol zavedený Wi-Fi Alianciou ako prechodné riešenie neistoty pri WEP. Wi-Fi Protected Access (WPA) je predchodca WPA2, namiesto implementácie úplného štandardu IEEE 802.11i implementuje iba 3. návrh tohoto štandardu, teda iba podmnožinu 802.11i. WPA2 používa blokovú šifru Advanced Encryption Standard (AES). Predchádzajúce WEP a WPA používajú prúdovú šifru RC4.

2.5.1 Typy zabezpečenia

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.

2.5.2 Šifrovacie algoritmy

TKIP

TKIP bol zostrojený tak aby sa dal jednoducho vložiť do nových firmvérov pre zariadenia siete 802.11. TKIP využíva rovnaký šifrovací algoritmus ako WEP. Štandardne však používa 128bitový kľúč a na rozdiel od WEP obsahuje dynamické dočasné kľúče. TKIP pracuje s automatickým kľúčovým mechanizmom, ktorý mení dočasný kľúč každých 10 000 paketov. Ďalšou veľkou

výhodou TKIP je Message Integrity Check (MIC), teda kontrola integrity správ. MIC je podstatne lepšie zabezpečenie integrity správ než dovtedy používaný jednoduchý kontrolný súčet CRC. MIC znemožňuje útočníkom zmeniť správy po prenose.

AES (CCMP)

Šifra využíva symetrický kľúč. Ten istý kľúč je použitý aj pre dešifrovanie. Dĺžka kľúča môže byť 128, 192 alebo 256 bitov. Metóda šifruje dáta postupne v blokoch s pevnou dĺžkou 128 bitov. Šifra sa vyznačuje vysokou rýchlosťou šifrovania. V súčasnej dobe nebol uverejnený známy prípad prelomenia tejto metódy ochrany dát.

EAP (Extensible Authentication Protocol)

Rozširujúci autentifikačný protokol. Protokol, pri ktorom sa na autentifikáciu používa digitálny certifikát, ktorý sa musí predinštalovať na klientský PC. Typicky sa používa s RADIUS serverom na autentifikáciu používateľov pri veľkých podnikových sieťach. EAP protokol je používaný v štandardoch 802.1X a v ochranách WPA Enterprise a WPA2 Enterprise.

LEAP (Lightweight EAP)

LEAP protokol vyvinula firma Cisco, je postavený na 802.1X a minimalizuje bezpečnostné chyby pri použití s WEP. Táto verzia EAP je bezpečnejšia ako EAP-MD5. Na autentizáciu používa MAC adresy. Samozrejme, že tento protokol nie je bezpečný pred crackermi. V súčasnosti firma CISCO odporúča používateľom aby používali novšie verzie EAP ako sú – EAP-FAST, PEAP, alebo EAP-TLS.

PEAP (Protected EAP)

PEAP nie je šifrovací protokol. PEAP len autentifikuje klientov v sieti. Na autentizáciu používa verejné kľúčové certifikáty. Potom sa vytvára šifrovaný SSL/TLS tunel medzi klientom a autentizačným serverom. Táto metóda bola vytvorená vďaka Cisco, Microsoft a RSA Security.

a) PEAPV0/EAP-MSCHAPV2

Je najčastejšie používaná forma PEAP. Vnútro autentizácie tvorí Microsoft's Challenge Handshake Authentication Protocol. PEAPv0/EAP-MSCHAPv2 je druhý široko podporovaný EAP štandard na svete.

b) PEAPV1/EAP-GTC

Bol vytvorený firmou Cisco. Microsoft nikdy nepridala do svojho operačného systému podporu pre PEAPV1. Hlavne aj preto je len zriedka používaný.

EAP-TLS (EAP – Transport Layer Security)

Bezpečnosť TLS (predtým oficiálne a teraz neoficiálne SSL – Secure Sockets Layer) protokolu je veľmi silná. Používa tzv. PKI (Public key infrastructure – infraštruktúru verejných kľúčov) na ochranu komunikácie pre RADIUS autentizačný server. Napriek tomu, že je tento protokol zriedka rozšírený, je považovaný ako jeden z najbezpečnejších štandardov EAP. Univerzálne podporovaný všetkými výrobcami wireless hardvéru a tiež Microsoftom.

EAP-TTLS/MSCHAPV2 (EAP – Tunneled Transport Layer Security)

Je to EAP protokol ktorý rozšíril EAP-TLS. Bol vytvorený firmami Funk Software a Certicom. Síce nie je natívna podpora od operačného systému Microsoft Windows, je široko podporovaný cez všetky platformy. Na používanie je potrebné nainštalovať malý program, napr. SecureW2.

EAP-TTLS ponúka veľmi dobrú ochranu. Klientský počítač nepotrebuje byť autentifikovaný cez certifikačnú autoritu - prihlásenie s PKI certifikátom na server, ale stačí klasické spojenie server – klient. Toto značne zjednodušilo procedúry nastavovania, pretože v tomto prípade nie je potrebné aby boli nainštalované certifikáty na každom klientovi.

EAP-SIM

Špecifický mechanizmus pre vzájomnú autentizáciu a prijatie kľúčového spojenia pri používaní GSM-SIM alebo GSM-based mobilných telefónnych sietí.

2.5.3 Výmena správ

IEEE 802.11i rozširuje IEEE 802.11-1999 tým, že poskytuje robustné zabezpečenie siete (RSN - Robust Security Network) s dvoma novými protokolmi: 4-Way Handshake a Group Key Handshake. Tie využívajú overovacie služby a kontrolu prístupových portov, sú opísané v IEEE 802.1X na vytvorenie a zmenu príslušných kryptografických kľúčov. RSN je bezpečnostná sieť, ktorá iba povoľuje tvorbu robustných bezpečnostných sieťových asociácií (RSNAs), ktoré sú typom asociácie používaným dvojicou staníc (STA) v prípade, že postup autentizácie alebo asociácie, medzi nimi zahŕňa 4-Way Handshake (4 cestný handshake).

Počiatočný proces overovania sa uskutočňuje, buď pomocou vopred zdieľanýého kľúča (PSK - pre-shared key), alebo na základe výmeny EAP prostredníctvom 802.1x (známy ako EAPOL, ktorý vyžaduje prítomnosť autentizačného servera). Tento proces zabezpečuje, že klientská stanica (STA) je overená prístupovým bodom (AP). Po PSK alebo autentizácii 802.1X, je generovaný zdieľaný tajný kľúč, nazvaný Pairwise Master Key (PMK). Pre odvodenie PMK z PSK, PSK je preložený cez PBKDF2-SHA1 ako kryptografická hashovacia funkcia. Ak sa vykonala výmena 802.1X EAP, PMK je odvodené od parametrov EAP vykonaných autentifikačným serverom.

Kontrola prístupu:

Domáce použitie:

- Zdieľaný kľúč ako v prípade WEP
- WPA-Personal, WPA2-Personal

Korporátne použitie:

- 802.1X (EAP, Radius)
- WPA-Enterprise, WPA2-Eterprise
- Výstupom autentizačného procesu je „Master key“ (MK) medzi stanicou a autentizačným serverom. Master key je platný len pre danú reláciu

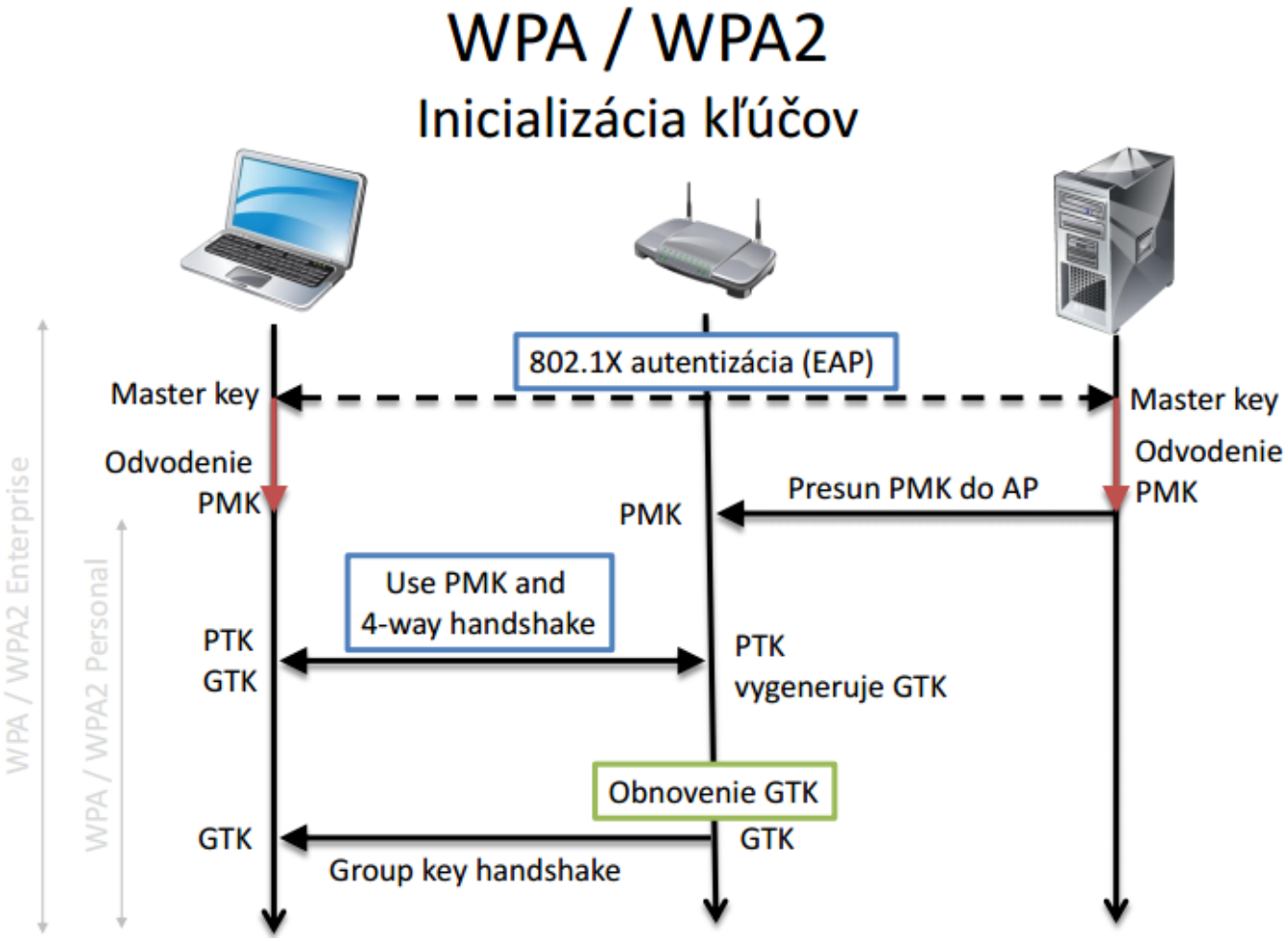
2.5.4 Inicializácia kľúčov

Automatická: využíva 802.1X

- Po skončení autentizácie 802.1X, stanica aj autentizačný server vypočítajú na základe Master Key tzv. Pairwise Master Key (PMK)
- Autentizačný server pošle PMK do AP
- Stanica a AP použijú PMK na vygenerovanie PTK (Pairwise transient key)
  - Na šifrovanie správ medzi AP a jednou STA
  - Vypočíta sa pomocou tzv. 4-way handshake protokolu
- Následne AP vygeneruje GTK
  - Na šifrovanie „broadcast“ správ od AP ku všetkým STA

Manuálna

- WPA/WPA2-Personal
- PMK je odvodené z pre-shared key, ďalej sa postupuje rovnako ako v predošlom bode



Obr.č.2 - *Inicializácia kľúčov* [1]

4-way handshake protokol

1. AP vysiela nonce hodnotu do STA (Anonce). Klient má teraz všetky atribúty pre zostrojenie PTK.
2. STA vysiela svoju vlastnú nonce hodnotu (SNonce) na AP spolu s MIC, vrátane autentifikácie, čo je v skutočnosti Message Authentication and Integrity Code (MAIC).
3. AP vytvorí a odošle GTK a poradové číslo spolu s inou MIC. Toto poradové číslo sa použije v budúcom multicast alebo broadcast rámci tak, že prijímacie STA môže vykonávať základnú detekciu.
4. STA pošle potvrdenie do prístupového bodu.

Group key handshake protokol

Skupinu dočaných kľúčov (GTK - Group Temporal Key), použitých v sieti môže byť potrebné aktualizovať kvôli uplynutiu prednastaveného časovača. Keď prístroj opustí sieť, GTK je potrebné

aktualizovať. Je to preto, aby sa zabránilo zariadeniu prijímať akékoľvek multicast alebo broadcast správy z AP.

1. AP vysiela novú GTK každému STA v sieti. GTK je zašifrovaný pomocou KEK, ktorý je priradený k tomuto STA, a chráni dáta pred manipuláciu, za použitia MIC.
2. STA potvrdzuje nový GTK a odpovedá na AP.

2.6 Štandard 802.11k – Assisted roaming

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.

2.7 Štandard 802.11r – Fast transition roaming

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

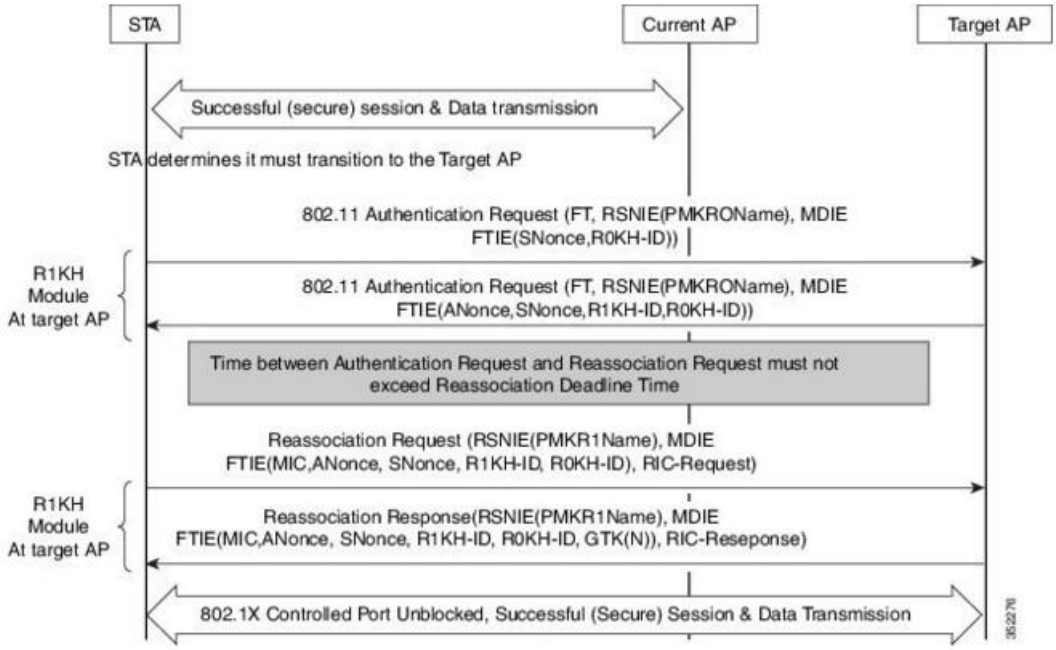
Handshake s novým AP sa vykoná pred samotným roamingom (FT - fast transition), teda mobilná stanica poskytne novému AP nevyhnutné informácie potrebné na prechod do tohto AP. Handshake

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**

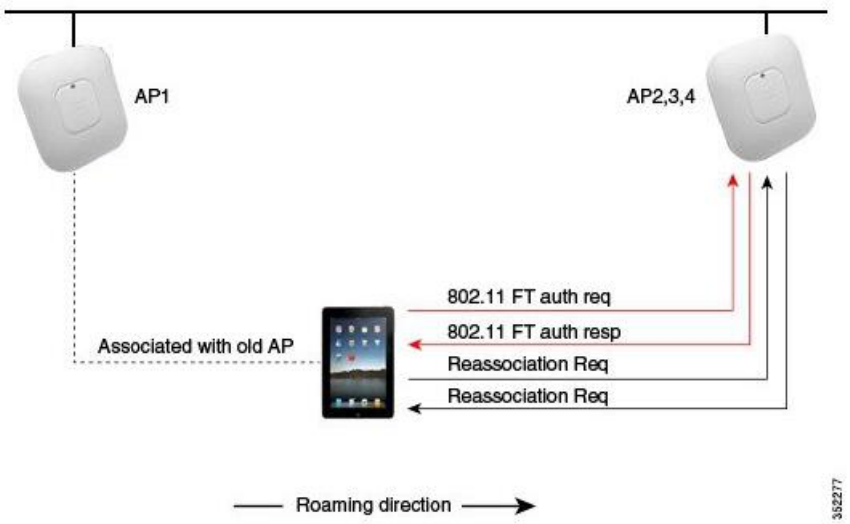
Klient komunikuje priamo s cieľovým AP pomocou IEEE 802.11 autentifikácie použitím FT autentifikačného algoritmu.



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

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

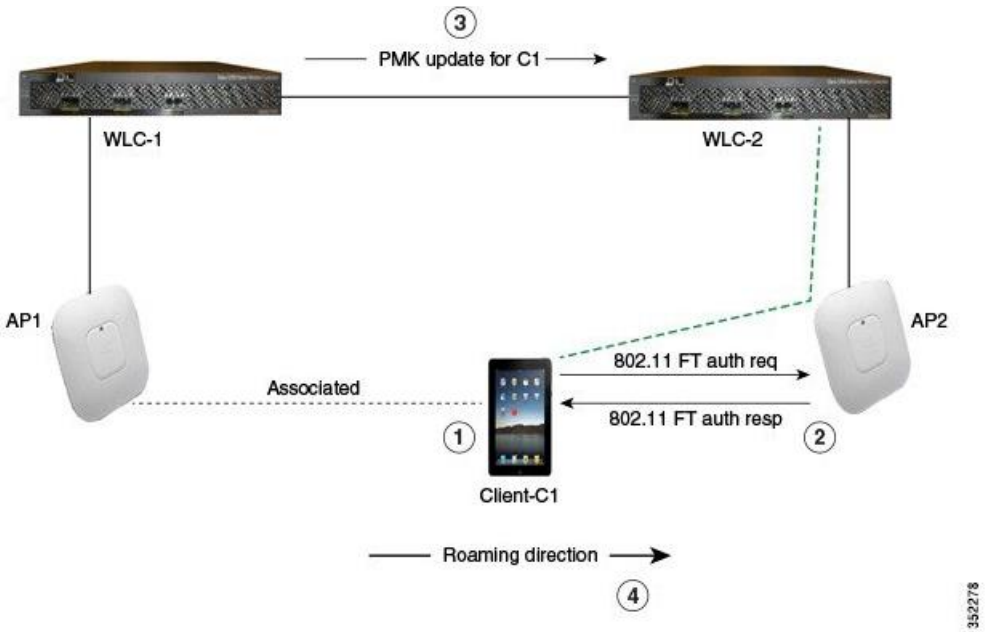
1. Klient je asociovaný s AP1 a chce prejsť do AP2.
2. Klient pošle na AP2 *FT authentication request* a potom od neho príjme *FT authentication response*.
3. Následne pošle na AP2 *reassociation request* a potom od neho príjme *reassociation response*.
4. Klient úspešne prejde do AP2.



Obr.č.4 - *Over the Air Intra Controller Roam* [4]

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

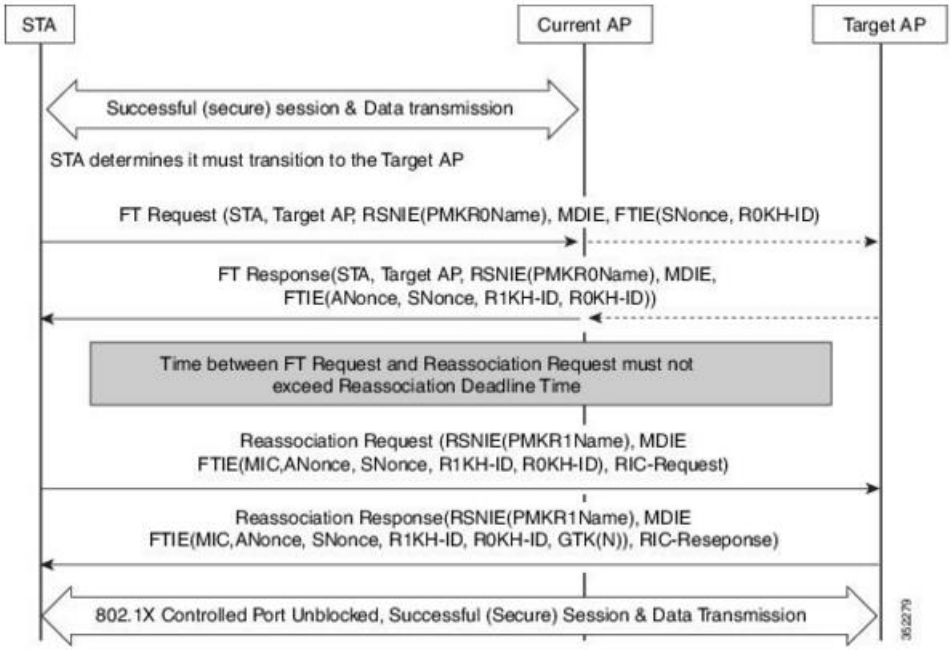
1. Klient je asociovaný s AP1 a chce prejsť do AP2.
2. Klient pošle na AP2 *FT authentication request* a potom od neho príjme *FT authentication response*.
3. Kontrolór, na ktorý je pripojený AP1 pošle *Pairwise Master Key* (PMK) na druhý kontrolór (na ktorom je pripojený AP2).
4. Klient úspešne prejde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



Obr.č.5 - *Over the Air Inter Controller Roam* [4]

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

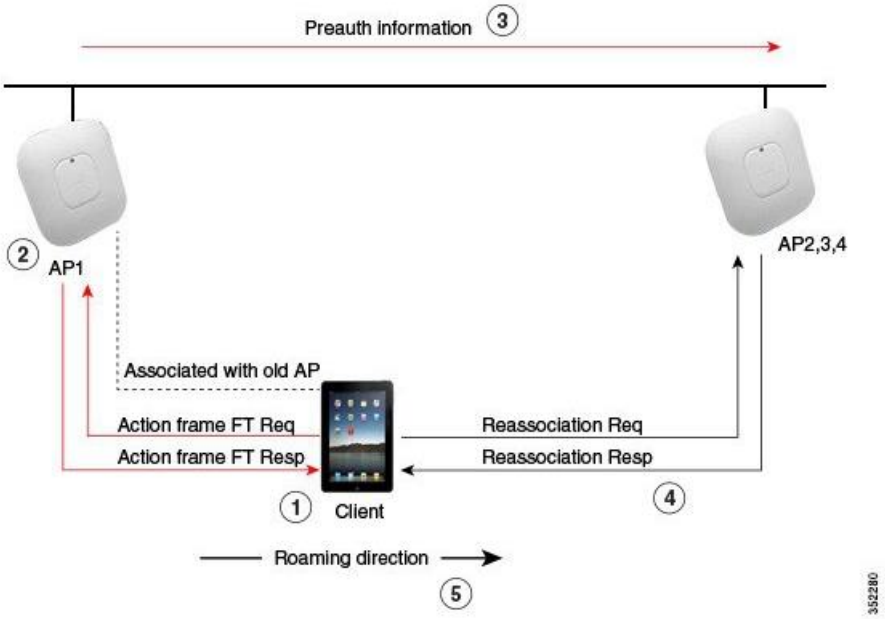
Klient komunikuje s cieľovým AP cez aktuálny AP. Komunikácia sa vykonáva prostredníctvom tzv. *FT action paketov* medzi klientom a aktuálnym AP a potom je poslaná cez kontrolór.



Obr.č.6 - *Over the DS Fast Transition roaming* [4]

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

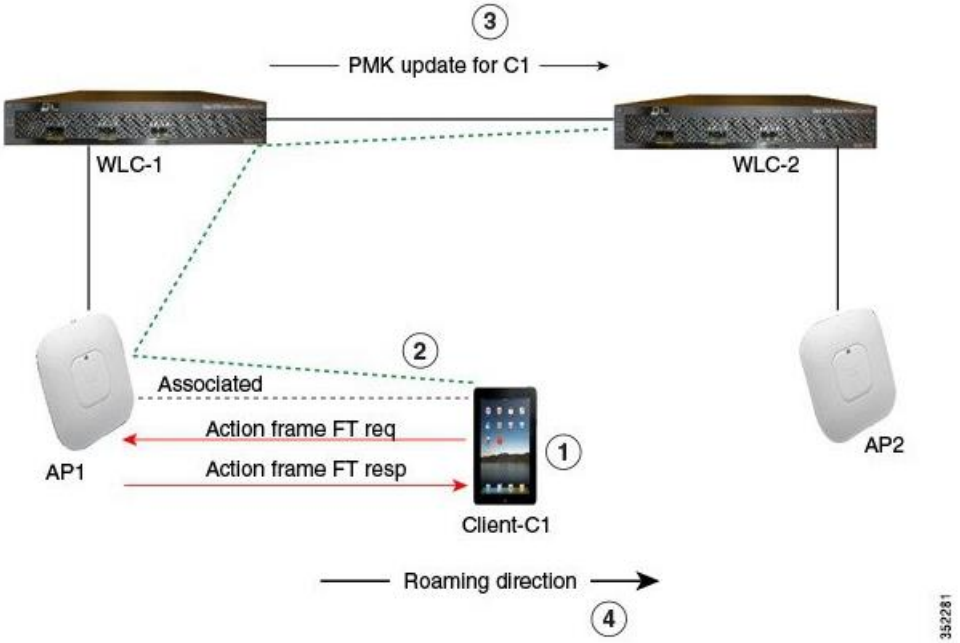
1. Klient je asociovaný s AP1 a chce prejsť do AP2.
2. Klient pošle na AP1 *FT authentication request* a potom od neho príjme *FT authentication response.*
3. Keďže sú oba AP pripojené na rovnaký kontrolór, pred-autentifikačné informácie sú poslané z kontrolóra na AP2.
4. Následne pošle klient na AP2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



Obr.č.7 - *Over the DS Intra Controller Roam* [4]

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.8 - *Over the DS Inter Controller Roam* [4]

2.8 SDN-Wifi

Wireless termination point (WTP) – predstavuje nám Access Point.

SD-RAN controller – predstavuje SDN kontrolér. Obsahuje viacero virtuálnych sietí alebo častí (sieťová časť je virtuálna sieť s SSID a setom WTP). Sieťové aplikácie nad kontrolórom, ktoré sú prístupné len pre sieťové časti, ktorým sú určené.

Používame 4 druhy abstrakcie a to:

1. **Light virtual access point (LVAP)** – Reprezentuje abstraktné spojenie medzi klientom a WTP. Každé fyzické WTP má uložené všetky LVAP, ktoré sú na ňom pripojené. Premiestnením LVAP z jedného WTP na druhé WTP dosiahneme efektívny handover. V podstate si každý klient vďaka LVAP myslí, že je v sieti sám, a to nám umožňuje, aby komunikovali WTP s klientom unicastom. LVAP obsahuje MAC adresu klienta, IP adresu klienta, lvap SSID a BSSID.
2. **Resource pool** – abstrakcia je silne spätá s abstrakciou LVAP. Obsahuje kolekciu zdrojov v čase, frekvencií a voľného miesta v sieti. Najmenšia jednotka je resource blok, ktorý je definovaný frekvenčným pásmom, časovým intervalom a WTP, na ktorom je prístupný. Na výber resource bloku, ktorý uspokojí požiadavky LVAP sa vyberá zo setu resource blokov z unionu voľných resource blokov podporovaných WTP. Samotný výber robí kontrolór.
3. **Channel quality and interference maps** – channel quality maps nám poskytujú celkový prehľad o sieti v podobe kvality kanálu vytvoreného medzi LVAP a WTP cez resource bloky. Na zistenie kvality kanálu ukladáme silu signálu pomocou RSSI. V rámci channel quality maps máme dve dátové štruktúry a to:

- User channel quality map (UCQM) – medzi LVAP a WTP.
- Network channel quality map (NCQM) – medzi dvoma WTP.

Obe sú trojdimenzionálne matice a v každej je záznam v decibeloch cez resource bloky.

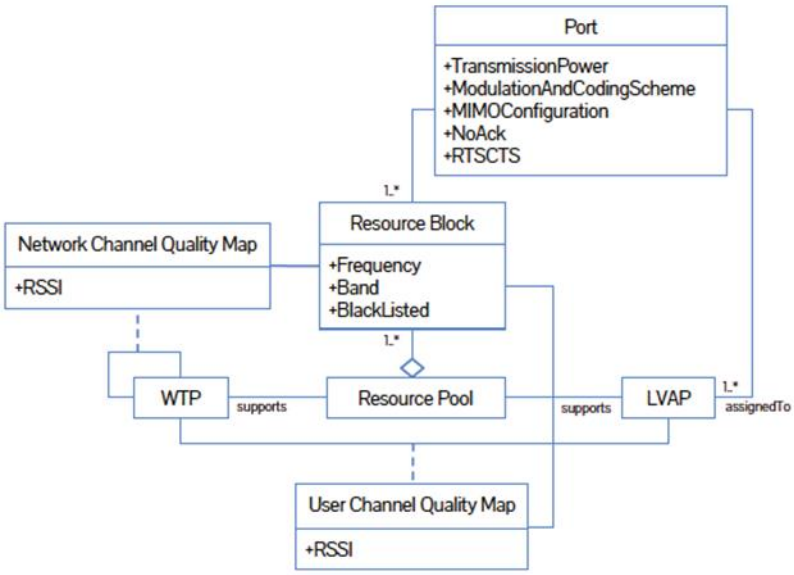
Port – dovoľuje kontrolóru prekonfigurovať alebo nahradiť určitú stratégiu ovládania medzi LVAP a WTP. Každé WTP má toľko portov koľko má na sebe pripojených LVAP. Samotný port obsahuje tri veci a to:

- p – sila prenosu,
- m – dostupné modulácie,
- a – MIMO konfiguráciu (počet priestorových prúdov).

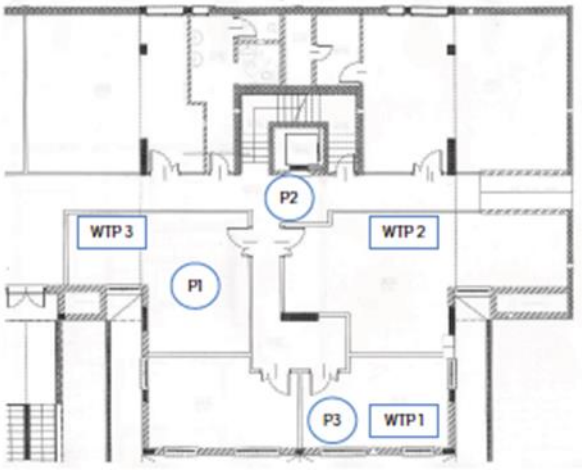
Prepojenie medzi jednotlivými abstrakciami môžeme vidieť na obrázku 9.

4. **Proximity detection** – jedno z možných využití tejto technológie. Ide o navigáciu vo vnútri budov, pretože GPS vo vnútorných priestoroch nedosahuje potrebnú kvalitu. Využívame

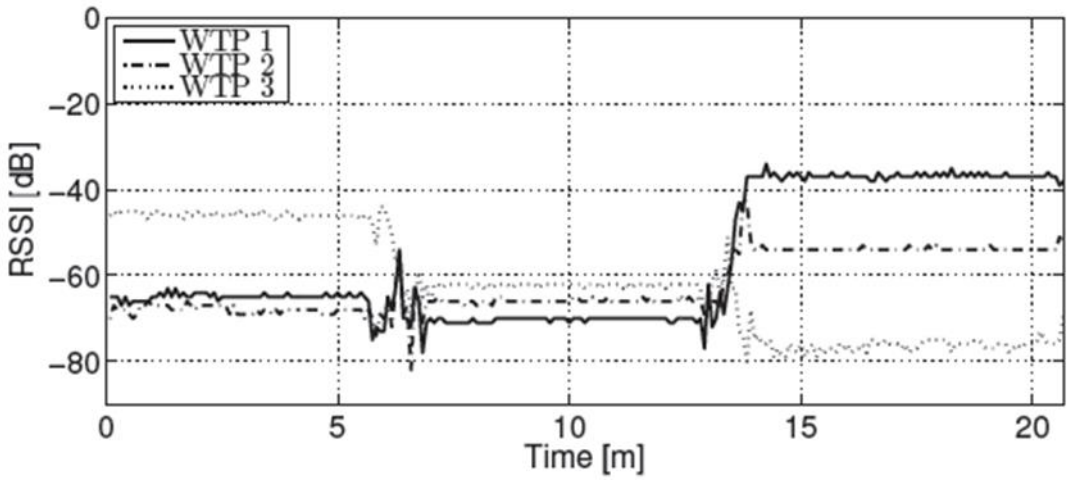
schopnosť vedieť o zariadeniach v blízkosti WTP pomocou RSSI stopovacej aplikácie. Na obrázku 10 môžeme vidieť rozmiestnenie jednotlivých WTP a pozícií testovacieho subjektu. Každých 5 minút stál subjekt na pozícií P a potom sa presunul na ďalšiu pozíciu. Výsledky testu môžeme vidieť na obrázku 11.



Obr.č.9 - *Diagram tried abstrakcií*



Obr.č.10 – *Rozmiestnenie v rámci poschodia*

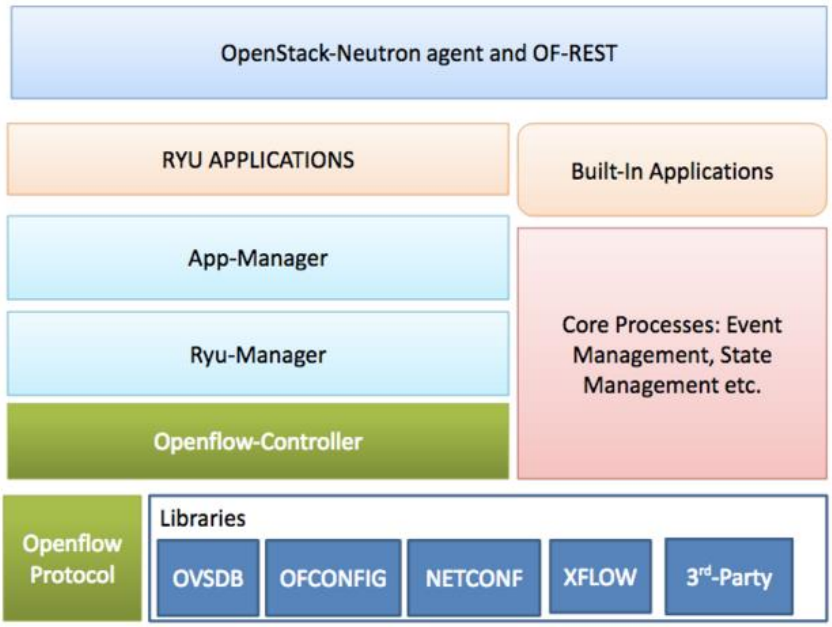


Obr.č.11 - *Výsledky testu*

2.9 SDN kontrolór RYU

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 12 je zobrazená architektúra Ryu.



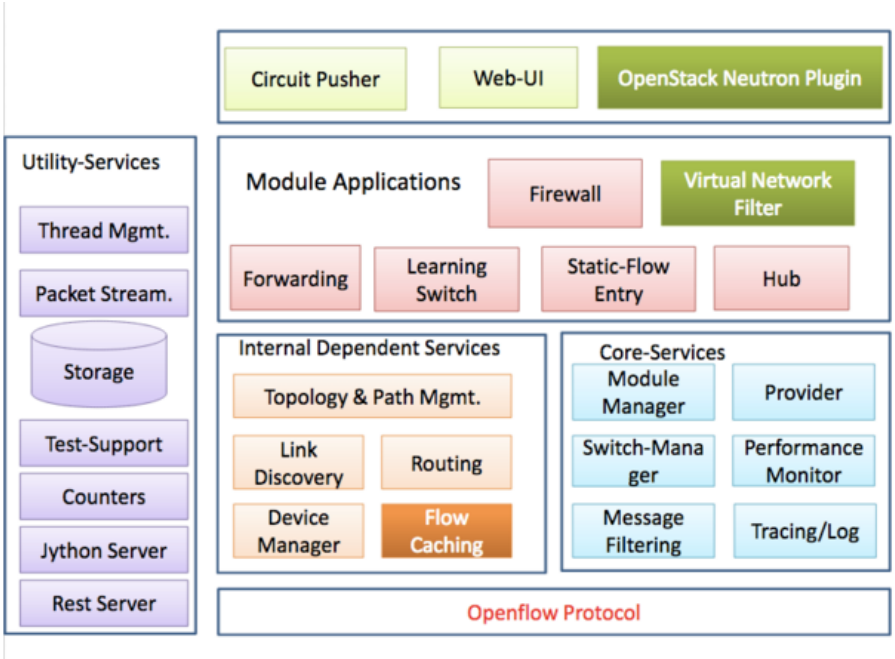
Obr.č.12 - *Architektúra Ryu* [2]

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.

2.10 SDN kontrolór Floodlight

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 13.



Obr.č.13 - *Architektúra Floodlight* [3]

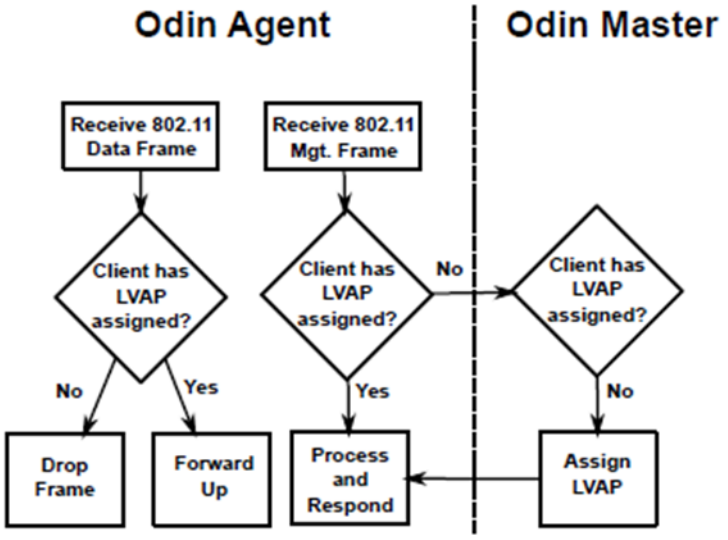
2.11 WLAN s Odin architektúrou

Light virtual AP (LVAP) – Reprezentuje abstraktné spojenie medzi klientom a AP. Každé fyzické AP má uložené všetky LVAP, ktoré sú na ňom pripojené. Premiestnením LVAP z jedného AP na druhé AP dosiahneme efektívny handover. V podstate si každý klient vďaka LVAP myslí, že je v sieti sám, a to nám umožňuje aby komunikovali AP s klientom unicastom. LVAP obsahuje MAC adresu klienta, IP adresu klienta, lvap SSID a BSSID.

Odin Master – V našom prípade openflow aplikácia nad kontrolórom. Je implementovaný nad Floodlight OpenFlow kontrolórom. Môže vytvoriť, pridať alebo odobrať LVAP a žiadať štatistiky z AP, updatovať jednotlivé forward tabuľky a podobne.

Odin Agent – Aplikácia nad fyzickým AP. Okrem pre SDN klasických forward tabuliek dokážu spracovať LVAP a ukladať informácie o klientoch ktorý sú naňho pripojený (používa tzv. radiotap headers).

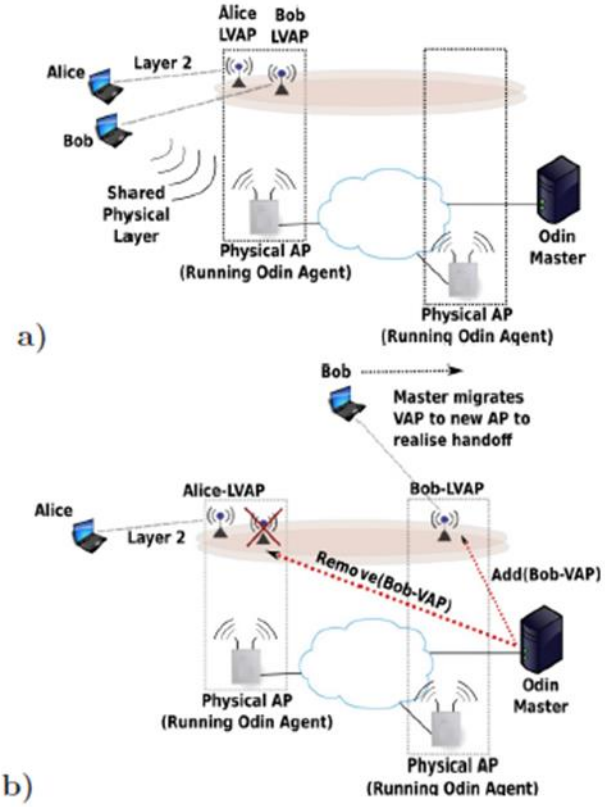
Ako to funguje – Odin agenti zachytávajú probe request od klientov. V prípade, že zachytí probe request správu od klienta, ktorého nepozná prepošle túto správu na Odin mastera. Pokiaľ pre tohto klienta ešte nebol vytvorený LVAP, tak ho master vytvorí a zapíše na agenta. Toto môžeme vidieť aj na obrázku 14.



Obr.č.14 – Zachytenie správy od klienta

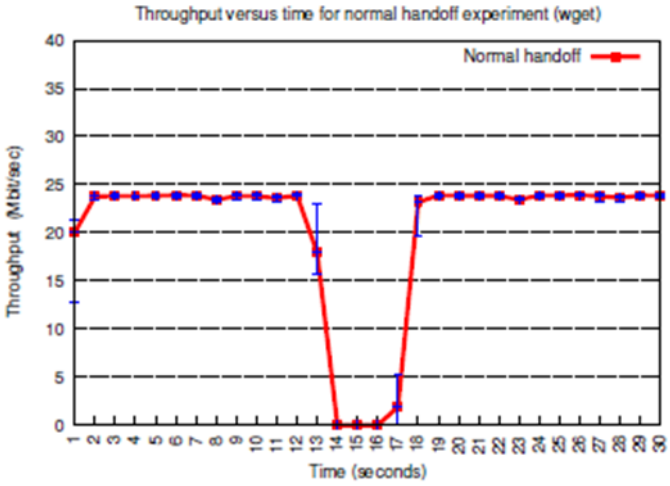
Agent potom odpovedá správou probe response s unikátnym BSSID, ktorý mu dodá master. Potom nasleduje autentifikácia a asociácia. Autentifikácia prebieha tak, že si agent uloží do LVAP kľúč na šifrovanie správ, na ktorom sa pri autentifikácií dohodne s klientom. Po asocíacii agent oznámi či bol povolený prístup do siete pre klienta, alebo nie.

Handover – prebieha tak, že master získava od agentov štatistiky o klientoch z probe správ, ktoré porovnáva so štatistikou od agenta na ktorom sú klienti pripojený. V prípade, že zistí že niekde má klient lepší signál, pošle správu agentovi, na ktorom je LVAP uložené na preposlanie tohto LVAP masterovi, ktorý ho pošle na druhé AP, ktoré má lepší signál, spolu so správou na následné zmazanie tohto LVAP, a agentovi, na ktorý sa má preposlať LVAP, aby pridal nové LVAP, ktoré mu master pošle, čo môžeme vidieť na obrázku 15.

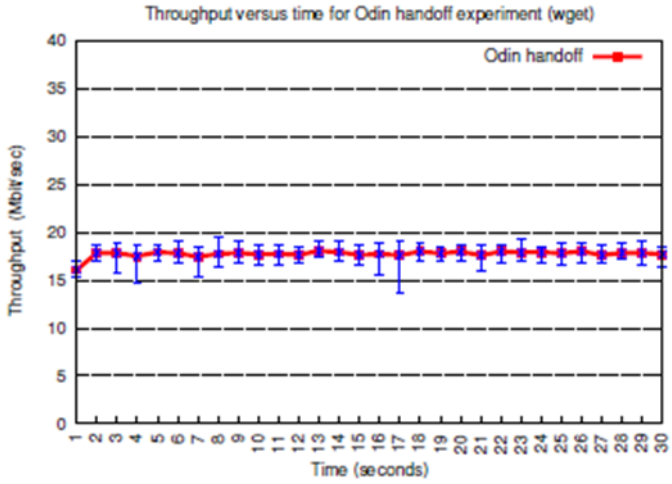


Obr.č.15 – Handover

Testovanie handoveru – na testovanie handoveru bola použitá sieť s jedným kontrolórom, dvoma AP a jedným používateľom. Pri teste boli sťahované dáta. Na obrázku 16 vidíme handover v klasickej sieti 802.11. Dôsledkom odpojenia a opätovného pripojenia do siete vidíme, že priepustnosť klesla. Na obrázku 17 môžeme vidieť handover v SDN sieti pri použití LVAP. Priepustnosť neklesne z dôvodu, že si jednotlivé AP iba vymenia LVAP daného používateľa bez toho, aby o tom vedel, pričom nedochádza k odpojeniu zo siete.



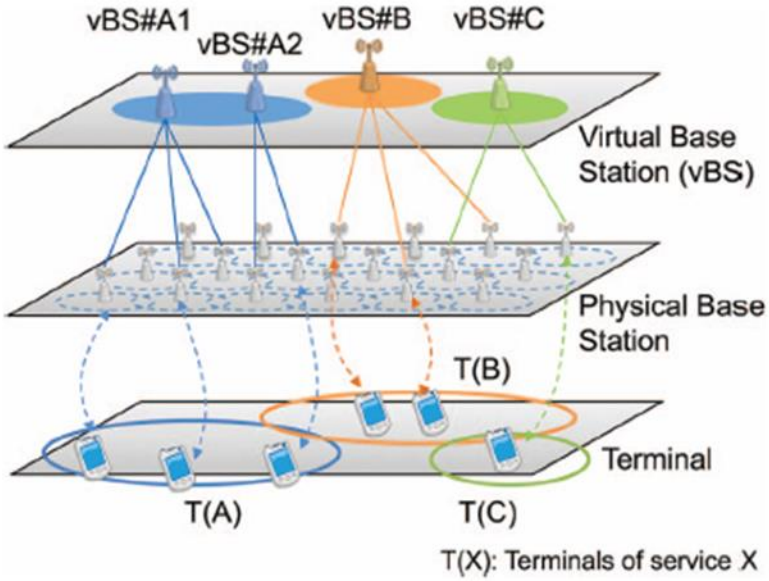
Obr.č.16 – 802.11 handoff



Obr.č.17 – LVAP handoff

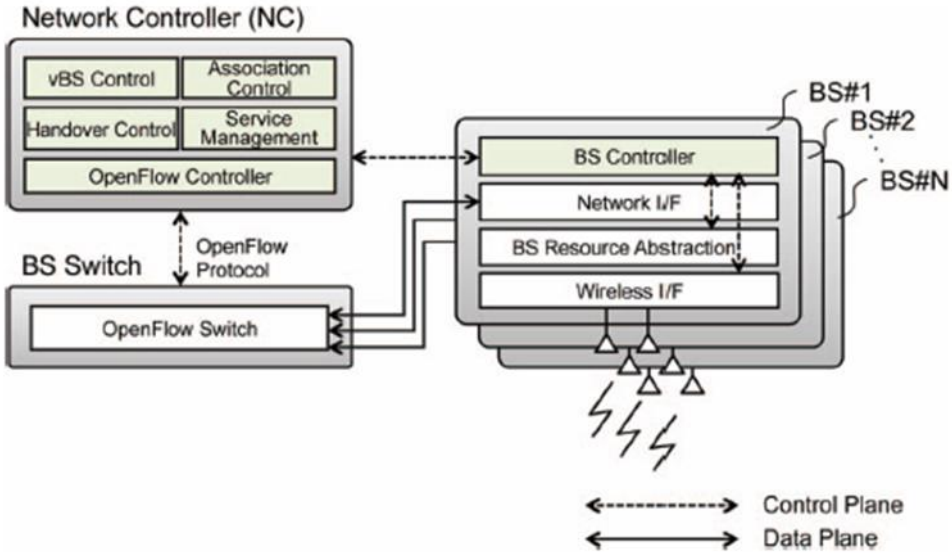
2.12 Virtualizácia Wifi siete

Virtualizácia Wifi siete – môžeme si ju predstaviť ako viacero fyzických AP, ktoré poskytujú jednu službu, zapojených do jedného virtuálneho AP. Na obrázku 18 môžeme vidieť viacero fyzických (na najnižšej vrstve) AP zapojených do viacerých virtuálnych AP (na najvyššej vrstve). Všetky virtuálne AP v rámci jednej siete majú rovnaké ESSID a BSSID. Na obrázku vidíme, že vBS#A1 a vBS#A2 patria do jednej siete.



Obr.č.18 – Model Wifi network virtualization

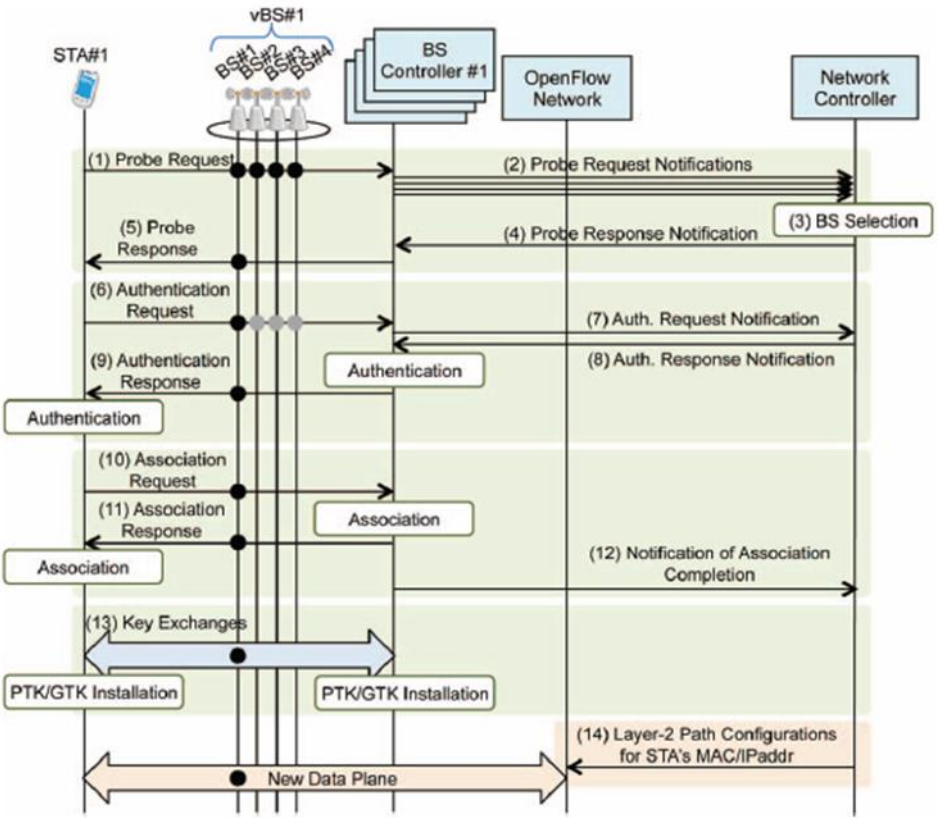
Funkčný model navrhnutej architektúry – Na obrázku 19 môžeme vidieť tento návrh, pričom sa skladá z kontrolóra, prepínaču a samotného zoznamu virtuálnych AP. Tento zoznam vytvára kontrolór, ako aj samotnú logiku asociácie spolu s kontrolórom BS pre jednotlivé BS, s ktorým potom komunikuje. Každé fyzické AP má rovnaké BSSID, ESSID a MAC adresu. Líšia sa len v kanále. Kontrolór vyberá, na ktoré fyzické AP sa používateľ pripojí podľa toho, koľko je na jednotlivých fyzických AP pripojených používateľov, pričom vyberie to s najmenším počtom.



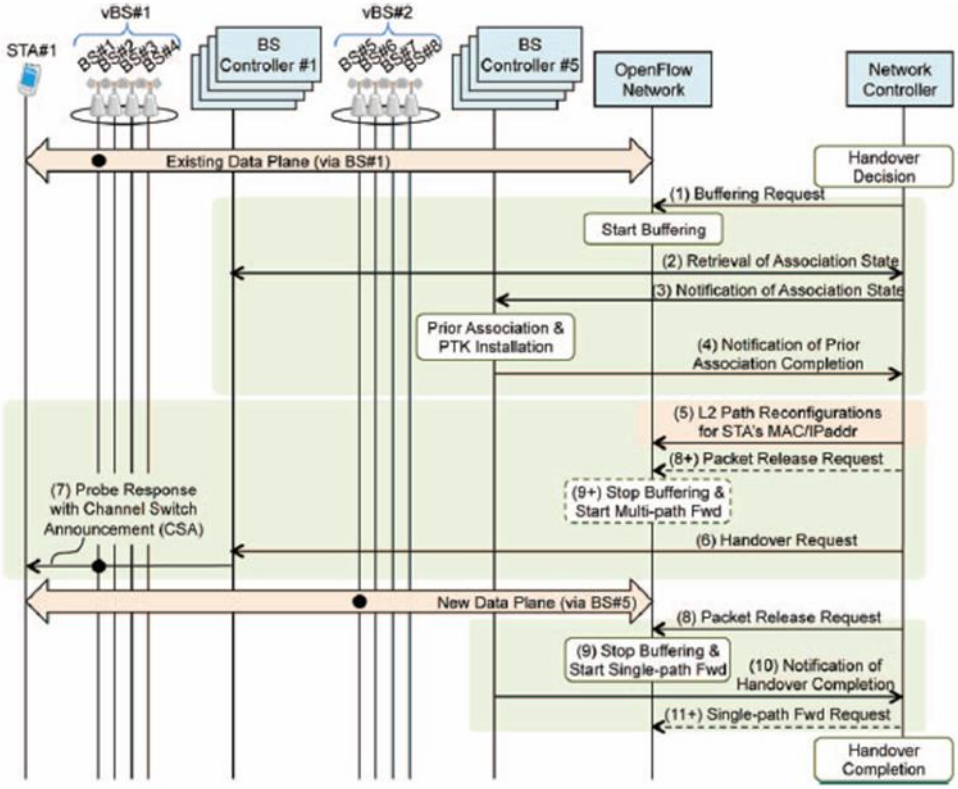
Obr.č.19 – Funkčný model navrhnutej architektúry

Pripojenie používateľa do siete – môžeme vidieť na obrázku 20. Používateľ (STA#1) posiela probe request správu do siete. Všetky fyzické AP ktoré túto správu dostanú ju prepošlú na BS kontrolór, ktorý následne prepošle všetky tieto requesty sieťovému kontrolóru. Ten následne vyberie jedno fyzické AP ktorému sa má priradiť daný používateľ. Nasleduje autentifikácia so sieťovým kontrolórom a následná asociácia, výmena kľúčom a následný data plane (na obrázku môžeme vidieť aj PTK ktoré je vytvorené 4-way handshakom medzi používateľom a BS). Celý proces pripojenia používateľa do siete vidíme prehľadne na obrázku 20.

Handover – vyvoláva sieťový kontrolór, ktorý aby sme zabránili strate údajov pošle openflow sieti aby začal pripravovať správy. Zo starého BS získa sieťový kontrolór stav a ten priradí novému BS. Následne si sieťový kontrolór prekonfiguruje cestu pre používateľa, a pošle starému BS správu Handover request, ktorý pošle používateľovi správu o zmene kanála. Potom kontrolór pošle openflow sieti nech prestane pripravovať správy a začne posielať správy cez nový BS používateľovi. Nový BS potvrdí handover sieťovému kontrolóru a tým sa skončil handover. Toto funguje pri jednocestnom móde. Máme však k dispozícii aj viaccestný mód. Ten funguje tak, že duplikuje pakety a preposiela ich do starého BS aj do nového BS zároveň, takže zariadenie môže hneď zmeniť BS. Po úspešnom handoveri sa znova prechádza na jednocestný mód. Celý priebeh handoveru môžeme vidieť na obrázku 21.



Obr.č.20 – Pripojenie používateľa do siete



Obr.č.21 – Priebeh handoveru

3 Návrh

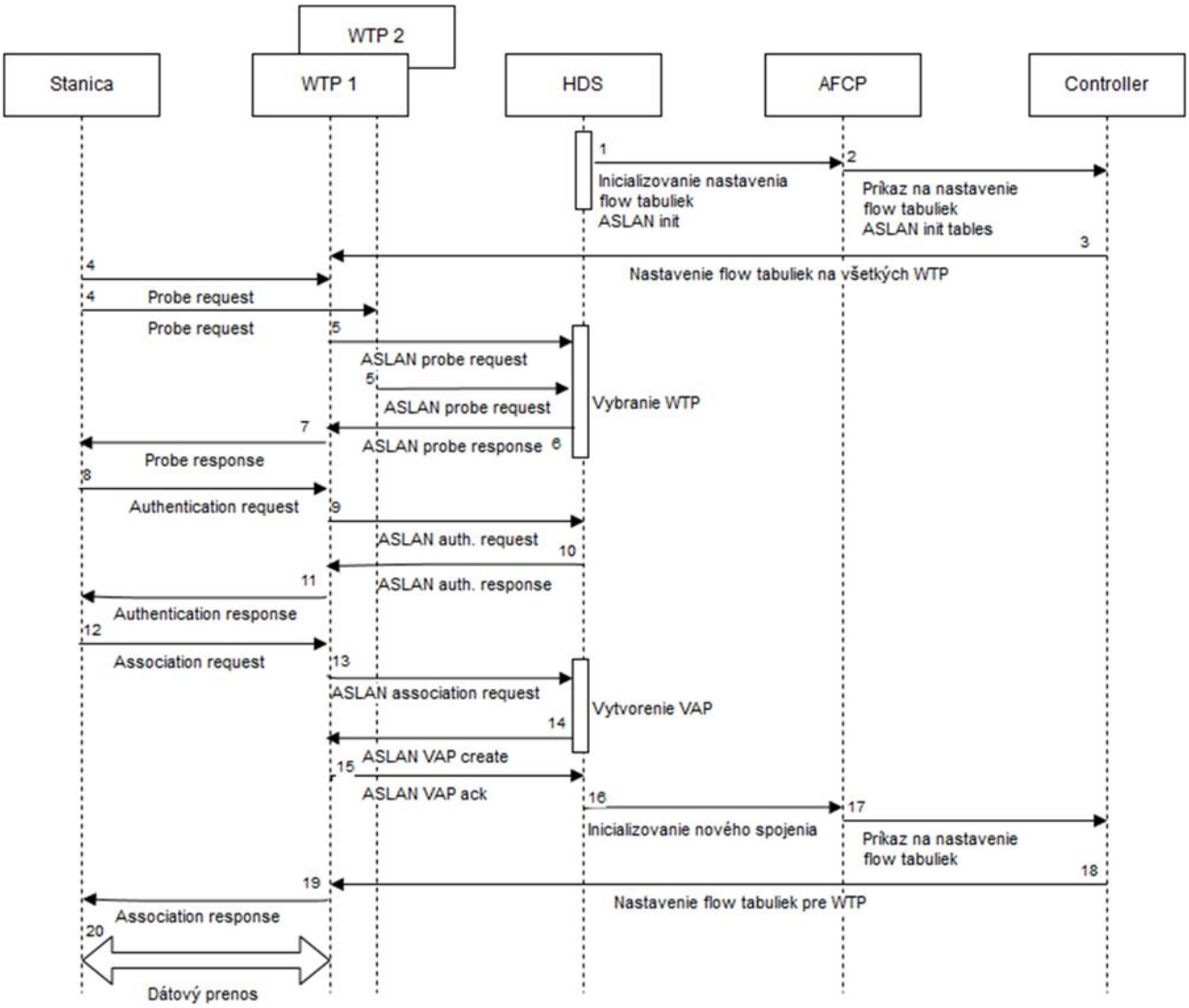
3.1 Komunikačný protokol

3.1.1 Scenár prihlásenia bez autentifikácie

1. Nastaviť open flow tabuľky pre probe správy – Ako prvé je potrebné nastaviť aby sa preposielali všetky probe správy na HDS server. Toto docielime nastavením pomocou AFCP kontrolór ktorý prepíše flow tabuľky na jednotlivých WTP.
    a. Nastavenie vyhľadávacieho pravidla na probe request správu (v 802.11 typ 00 čiže mgmt a subtype 0100).
    b. Nastavenie akcie na preposlanie cez port ktorým sme pripojený k danému WTP (napríklad output:2 čím nám WTP prepošle správu na port 2).
2. Nastaviť open flow tabuľky pre autentifikačné správy – Nastavujú sa rovnako ako pri probe správe len s rozdielom v prvom kroku – namiesto 0100 sa použije 1011. Keďže používame nezabezpečený štýl pripojenia, neoverujú sa žiadne kľúče.
3. Nastavenie open flow tabuľky pre potvrdenie autorizácie – Opäť prebieha rovnako ako pri vyššie spísaných správach, až na kód, ktorý je v tomto prípade 0000 pre Association request. V prípade ze prebehne asociácia v poriadku a stanica sa môže pripojiť do siete, uloží si HDS server pre dané VAP na ktorom fyzickom WTP je stanica pripojená.
4. Prijatie probe request správy od WTP– V prípade že sa chce stanica pripojiť do siete pošle probe request správu, pomocou ktorej zisťuje dostupné siete. Ak WTP zachytí probe request správu, prepošle ju podľa vopred stanoveného pravidla vo flow tabuľkách na HDS server.
5. Výber WTP ktorý má komunikovať so zariadením – Po zachytení probe správy na HDS servery musí byť vybrané jedno WTP ktoré bude komunikovať so zariadením. Po výbere daného WTP sa mu pošle VAP pre stanicu s ktorým má komunikovať. Komunikácia pritom prebieha unicastovo.
6. Poslanie ACK pri asociácií HDS serveru – Po úspešnom prebehnutí asociácie sa posiela HDS serveru potvrdenie. V prípade že dostane po autentifikácií WTP od zariadenia správu Association request prepošle HDS serveru túto správu.
7. Vytvorenie virtuálneho AP(VAP) – keďže používame sieť bez zabezpečenia, nemá zmysel čakať s vytvorením VAP na samotné wep/wpa správy. V prípade že by sme použili wep/wpa, museli by sme čakať na priebeh autentifikácie keďže tá prebieha cez HDS server v ktorom sa uložia autentifikačné informácie aby nemuselo dochádzať k odpojeniu

používateľa od siete a opätovnému pripojeniu. Následne po vytvorení VAP WTP pošle HDS serveru správu ASLAN VAP ack o vytvorení VAP.

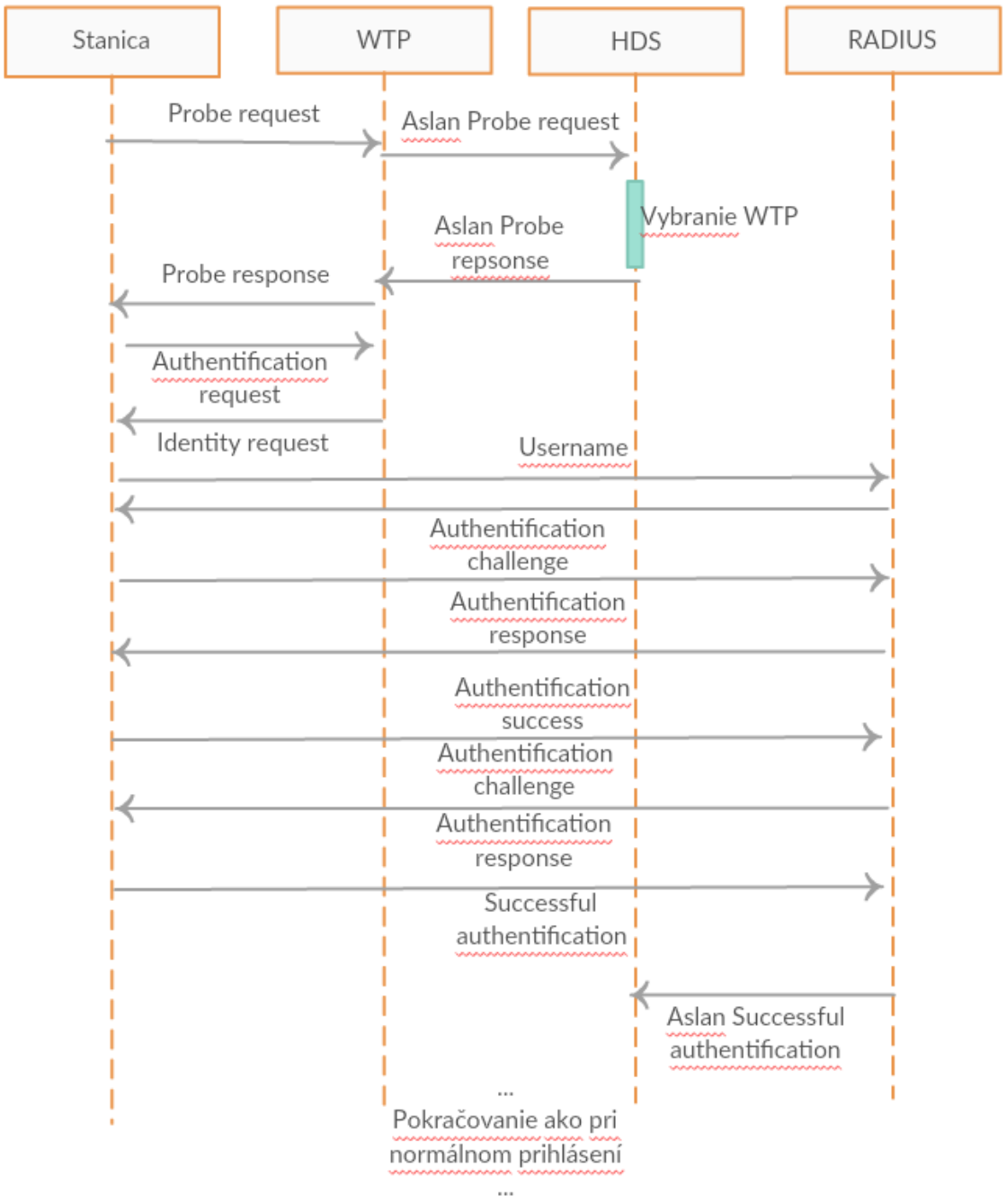
8. Nastavenie cesty cez flow tabuľky pomocou kontrolóra cez AFCP.
9. Po asociácií prebieha následne dátový prenos.



Obr.č.22 – *Výmena správ pri prihlasovaní*

3.1.2 Scenár prihlásenia s autentifikáciou pre RADIUS server

1. Stanica pošle Probe request do WTP.
2. WTP odošle ASLAN Probe request do HDS.
3. HDS sa rozhodne, ktoré WTP sa so stanicou spojí (určujúci faktor je najlepšia kvalita signálu).
4. HDS odošle odpoveď na WTP ASLAN Probe response.
5. Probe response sa pošle z WTP stanici.
6. Stanica požiada o autentifikáciu (Authentification request).
7. WTP požiada stanicu o identitu (Identity request)
8. Stanica pošle username, ktoré sa cez WTP prepošle do RADIUS server.
9. RADIUS prepošle cez WTP Authentification challenge.
10. Stanica prepošle cez WTP Authetification response.
11. RADIUS server prepošle cez WTP Authentification success.
12. Stanica pošle cez WTP Authentification challenge do RADIUS server.
13. RADIUS cez WTP odošle Authentification response.
14. Stanica odošle cez WTP Successful authentification do RADIUS server.
15. RADIUS server odošle Aslan Successful authentification na HDS server.
16. Pokračuje sa ako v scenári prihlásenia.

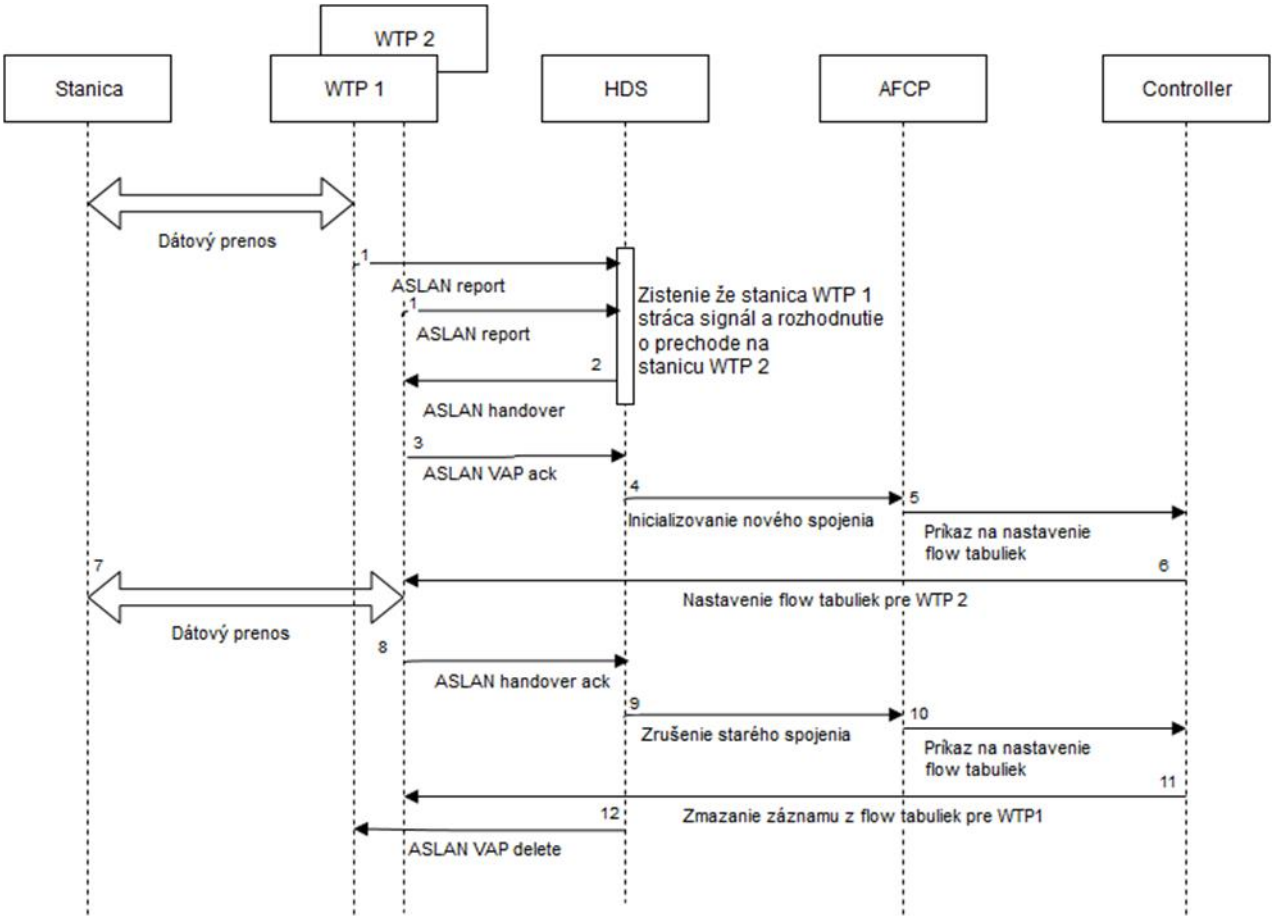


Obr.č.23 – *Výmena správ pri autentifikácii*

3.1.3 Scenár reasociácia

1. Zistenie strácajúceho sa signálu – Z každého WTP dostáva periodicky HDS server správy o sile signálu voči zariadeniam. HDS server zistí z periodických správ od WTP že na inom WTP dosahuje lepší signál.

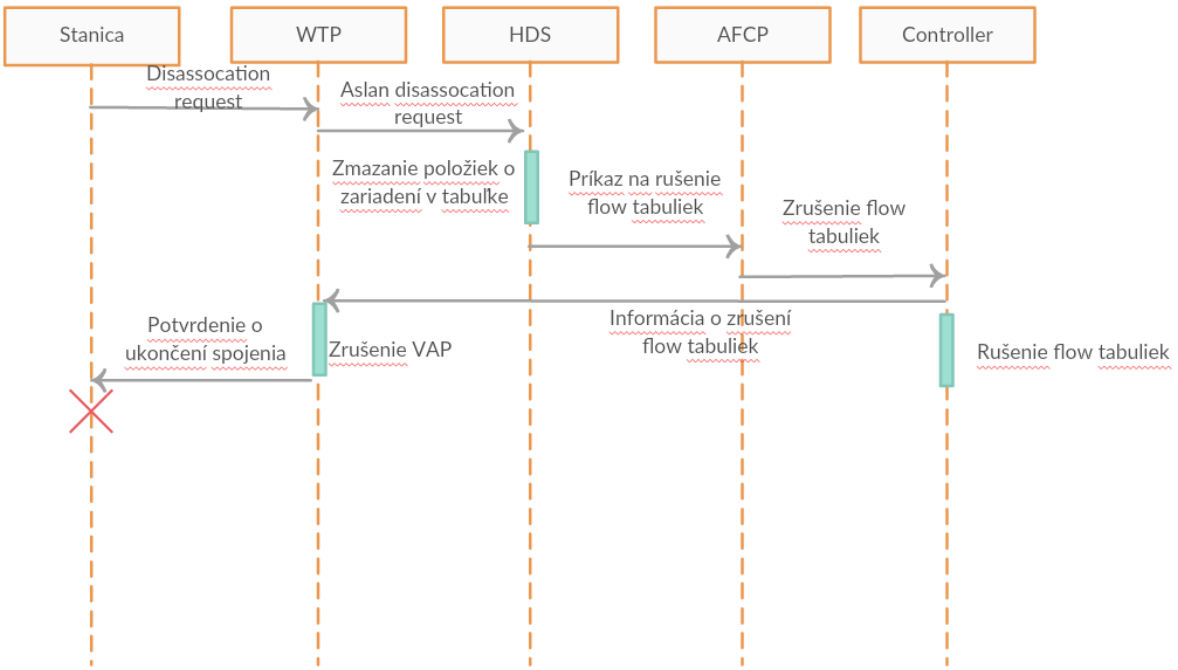
2. Rozhodnutie o WTP na ktorý sa má robiť reasociácia – V prípade že máme v sieti WTP s rovnakým kanálom, tak sa na chvíľu zosilní signál na tomto WTP na čas reasociácie. V inom prípade vyberáme WTP ktoré má najväčšiu silu signálu.
3. Preposlanie VAP novému WTP – Novému WTP sa prepošle VAP daného zariadenia. Nové WTP si ho uloží.
4. Nastavenie flow tabuliek na novom WTP – HDS server cez AFCP nastaví pomocou kontroléra flow tabuľky na novom WTP.
5. Oboznámenie zariadenia so zmenou kanála – V prípade že je nutná zmena kanála sa oboznámi stanicu o zmene kanála na ktorom má komunikovať s WTP.
6. Potvrdenie prepojenia nového WTP so zariadením – V prípade, že sa stanica úspešne spojí s novým WTP, pošle nové WTP HDS serveru potvrdenie o úspešnom prechode zariadenia.
7. Zmazanie záznamu flow tabuliek na starom WTP - V prípade, že prebehla reasociácia úspešne, HDS server pošle správu AFCP aby pomocou kontroléra zmazal na starom WTP záznamy vo flow tabuľkách o stanici.
8. Zrušenie VAP na starom WTP – V prípade, že prebehla reasociácia úspešne, pošle HDS server starému WTP správu na zrušenie VAP zariadenia.



Obr.č.24 – *Výmena správ pri reasociácii*

3.1.4 Scenár odhlásenia:

1. Odhlásenie – WTP odošle informácie o zariadení, že chce ukončiť komunikáciu.
2. Správa odhlásenia – odosiela sa cez WTP na HDS server. Správa je typu mgmt a podtyp je 1010 čo znamená Disasociácia. Tento paket je jednocestná komunikácia (čo znamená, že nepotrebuje potvrdenie, alebo odozvu) a musí byť akceptovaná. Takáto správa učiní efekt ihneď po prijatí.
3. Prijatie správy odhlásenia – HDS príjme správu, zmaže položku o zariadení vo svojej tabuľke.
4. Zrušenie ciest – na AFCP sa odošle správa, aby sa zrušili cesty cez flow tabuľky do VAP zariadenia, ktoré sa odhlasuje.
5. Zruší sa VAP a komunikácia je zrušená.



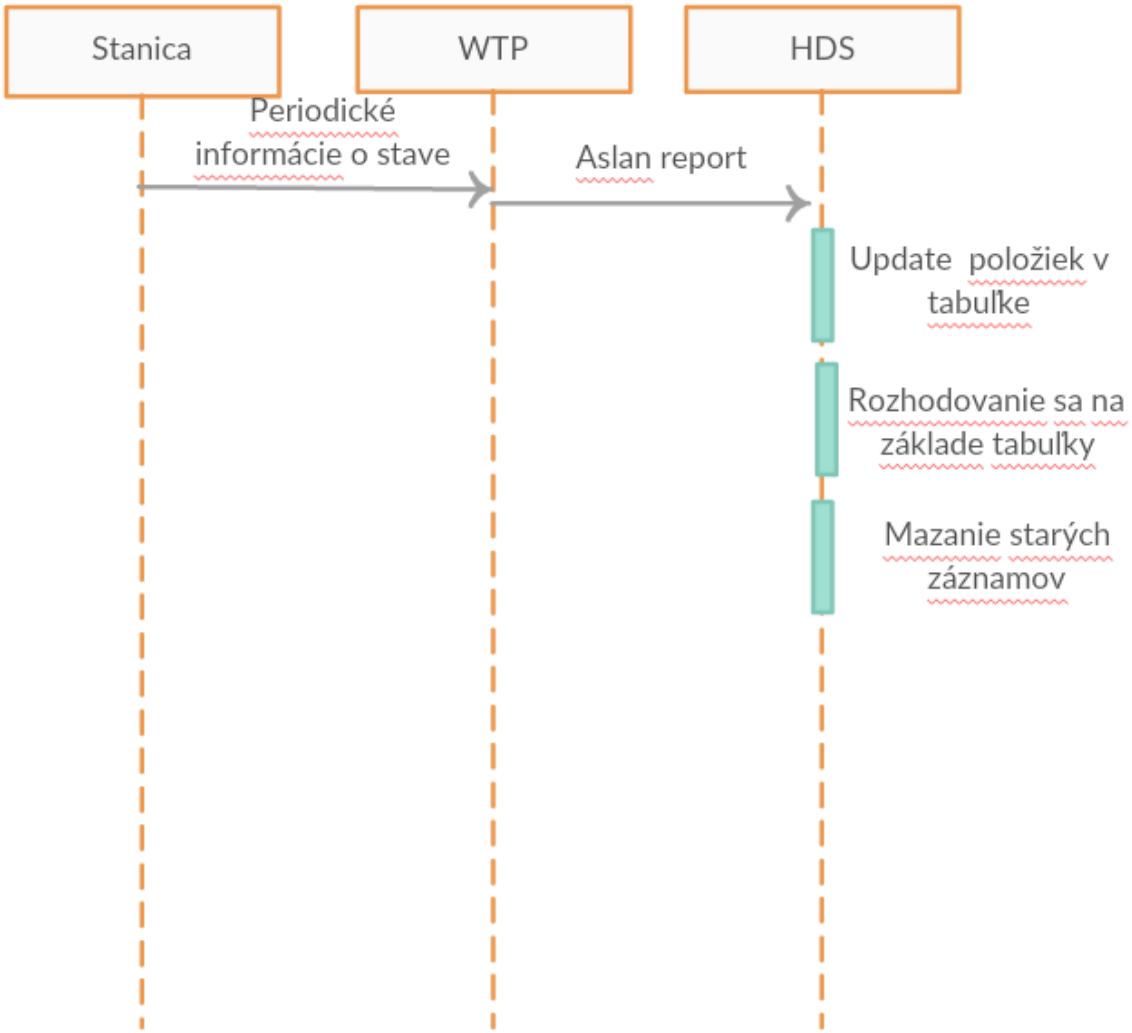
Obr.č.25 – *Výmena správ pri odhlásení*

3.1.5 Scenár monitorovania

1. Monitorovanie – WTP monitoruje všetky zariadenia v sieti.
2. Posielanie stavových správ - Všetky WTP periodicky posielajú správy o stave signálu voči zariadeniam v sieti. Tieto správy sa posielajú HDS serveru.
3. Ukladanie stavov – HDS server všetky prijaté správy o stave signálu porovnáva. Uchováva si vždy posledných 5 stavov z WTP ku ktorému je daná stanica pripojená.
4. Rozhodnutie o zmene WTP – V prípade že už signál z WTP ku ktorému je pripojené zaradenie slabne (posledných 5 stavov ukazujú pokles signálu) a HDS zistí že je v sieti iné WTP ktoré má za posledných 5 stavov lepšiu intenzitu signálu, rozhodne sa HDS o handovery na nové WTP.

3.1.6 Scenár reportovania:

1. Reportovanie – WTP odosiela informácie koľko má zariadení a aká je sila signálu jednotlivých zariadení.
2. Odosielanie reportovacích správ – WTP periodicky odosielajú správy HDS serveru o stave pripojených zariadení.
3. Tabuľka stavov – prijaté stavy na HDS servery sa ukladajú do tabuliek, ktoré majú 30 sekundové časovače životnosti.
4. Rozhodovanie – HDS sa rozhoduje podľa informácií či spraviť prechod.

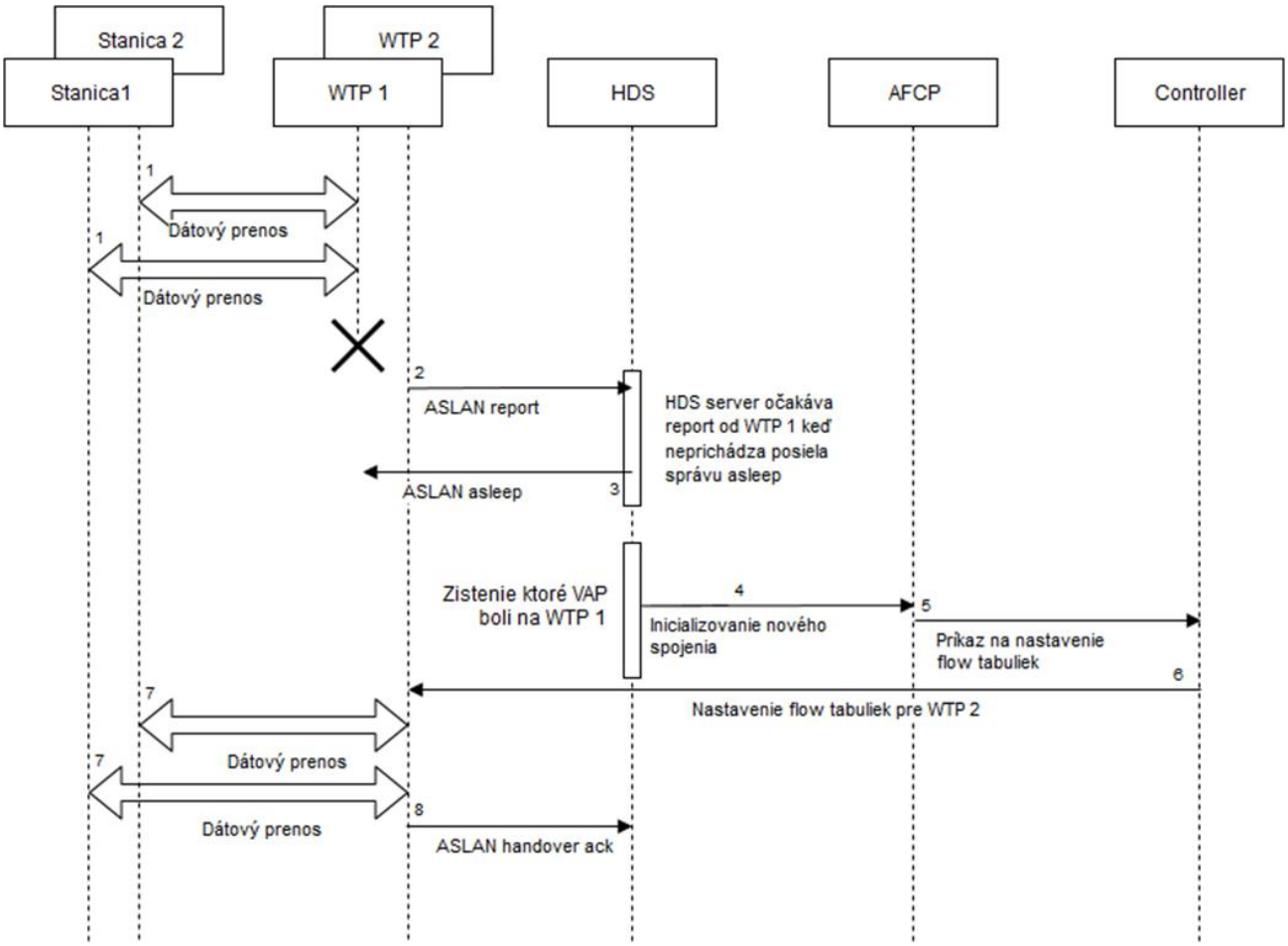


Obr.č.26 – *Výmena správ pri reportovaní*

3.1.7 Scenár pád WTP

1. Zistenie pádu WTP – HDS server očakáva periodicky hlásenie od WTP v sieti. V prípade, že sa WTP nehlási pošle mu HDS server správu (ASLAN asleep). Ak na túto správu WTP neodpovie nastal pád stanice.
2. Zistenie všetkých VAP ktoré boli pripojené na WTP - HDS server pri páde stanice musí okamžite reagovať, a snaží sa nájsť vhodné WTP pre každú stanicu, ktorá bola pripojená na nereagujúce WTP. Ako prvé sa hľadajú WTP ktoré boli v rámci jedného kanála. Ak takéto WTP nie je k dispozícií snaží sa pripojiť stanicu k WTP ktoré je ako druhé v tabuľke signálov.
3. Presmerovanie staníc na WTP – HDS server pošle správu novému WTP na vytvorenie všetkých VAP ktoré majú byť pripojené z nereagujúceho WTP.

4. Nastavenie flow tabuliek na novom WTP – HDS server štandardnou cestou nastaví flow tabuľky na novom WTP.
5. Prenos dát.



Obr.č.27 – *Výmena správ pri páde WTP*

3.1.8 Návrh protokolu

V prípade správy ASLAN VAP create používanej na vytvorenie VAP na danom WTP používame vopred stanovený protokol, ktorý vidíme na obrázku vyššie. V prípade ostatných správ používame protokol, ktorý vidíme na obrázku nižšie.

ASLAN VAP create

Message type 1B	MAC addr. 6B	IPv4 addr. 4B	VAP BSSID 6B	VAP SSID 32B	Data 464B

ASLAN other messages

Message type 1B	Data 512B

Obr.č.28 – *Návrh VAP a správ*

3.2 WTP - Wireless Termination Point

3.2.1 Personal AP

Personal AP je experimentálny projekt, ktorý prináša výrazné zlepšenia do problematiky rýchleho a plynulého prechodu medzi prístupovými bodmi. Autori projektu odstránili nutnosť reasociácie klienta pri roamingu z jedného prístupového bodu na druhý.

Klasické prístupové body sú rozdelené na dva základné typy – **WTP** a AC. AC funguje ako centrálny riadiaci bod celého systému, prípadne podsystému ak sa v ňom nachádza viacero takýchto radiacich bodov. **WTP** predstavuje prístupový bod, ktorý má implementovanú väčšinu funkcií klasického prístupového bodu používaného v bezdrôtových systémoch. Takýto spôsob rozdelenia funkcionality sa nazýva *„Local MAC“*.

Okrem spomenutého spôsobu existujú aj ďalšie: *„Split MAC“*, ktorý sa vyznačuje tým, že na **WTP** zariadení sú implementované iba funkcie, na vykonanie ktorých je potrebné minimálne oneskorenie a *„Remote MAC“*, kde všetka funkčnosť prístupového bodu je implementovaná na strane centrálneho riadiaceho bodu AC.

Hlavnou myšlienkou celého tohto spôsobu riadenia prevádzky v sieti je, že každý klient pripojený do systému má pridelený vlastný „virtuálny― prístupový bod. Všetky prístupové body v sieti pravidelne posielajú hlásenia o svojich pripojených klientoch. Ak nastane situácia, že je vhodné vykonať roaming mobilného účastníka na nový prístupový bod, starý prístupový bod odošle všetky informácie o účastníkovi novému. Následne nový prístupový bod takmer ihneď začne pracovať rovnako ako starý, t.j. s rovnakým BSSID a pod. Výhodou je, že mobilný klient nepocíti žiadnu zmenu. To znamená, že každý klient má v podstate svoj vlastný „virtuálny― prístupový bod, s ktorým je asociovaný a ktorý si **WTP** zariadenia dokážu medzi sebou posielať. Veľkou výhodou Personal AP je, že nepotrebuje žiadne dodatočné zmeny na strane účastníka.

3.2.2 Remote MAC

Rozdelenie *Remote MAC* je charakteristické tým, že všetky funkčné prvky sú implementované na centrálnom bode riadenia AC.

WTP zariadenia slúžia v podstate len na preposielanie dát AC. Ich ďalšou funkciou je zbieranie informácií o účastníkoch pohybujúcich sa v sieti. Tieto informácie následne posielajú pomocou navrhnutého protokolu centrálnemu riadiacemu bodu AC. Na základe týchto informácií si AC vytvára graf susedov, ktorý mu slúži na rozhodovanie, ktoré WTP zariadenie použiť pri roamingu.

WTP zariadenia tak zohrávajú dôležitú úlohu v navrhovanej architektúre. Predstavujú prepojovací systém medzi riadiacou jednotkou a mobilnými klientskymi stanicami. Zabezpečujú ich vzájomnú

komunikáciu a riadia prístup do siete. Zároveň prepájajú dve sieťové architektúry a to bezdrôtový systém štandardu 802.11 a pevnú sieť štandardu 802.3.

3.2.3 Virtual AP

Myšlienkou navrhovaného spôsobu riadenia prevádzky v sieti je, že každý klient pripojený do systému má pridelený vlastný „virtuálny“ prístupový bod. Jedinečnosť prístupového bodu je určená na základe BSSID, ktorý predstavuje adresu prístupového bodu. Keďže každá sieťová karta má k dispozícií iba jednu MAC adresu, je teda potrebné vytvoriť virtuálne rozhrania pre tieto účely. Tým zabezpečíme dostatok komunikačných adries pre všetkých pripojených klientov. BSSID, ktoré je pridelené mobilnej stanici, využíva stanica od prvého prihlásenia do siete až po odpojenie a teda v rámci mobility adresa „cestuje s ňou“.

Príklady názvov rozhraní v prostredí Linux:
wl0 – predstavuje fyzické rozhranie
wl0.1 – predstavuje virtuálne rozhranie s ID 1 nad fyzickým rozhraním wl0
wl0.2 – predstavuje virtuálne rozhranie s ID 2 nad fyzickým rozhraním wl0

Virtuálne rozhrania môžu mať odlišnú MAC adresu, ale nemôžu vysielať na inom vysielacom kanáli ako je nastavené fyzické rozhranie patriace k danému virtuálnemu rozhraniu.

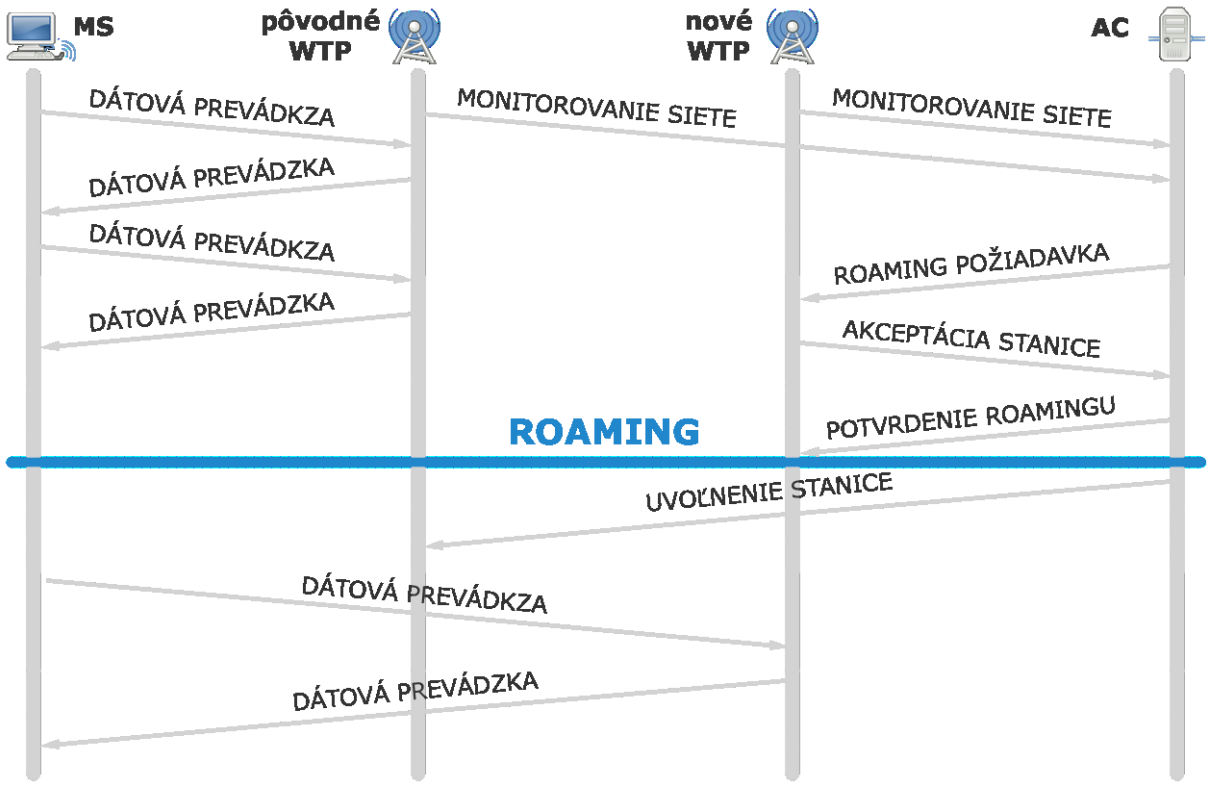
3.2.4 Monitoring na WTP

Každé WTP zariadenie si uchováva informácie o všetkých staniciach v jej dosahu. Tieto údaje získava z monitorovania bezdrôtovej siete. Konkrétne sa jedná o riadiace rámce 802.11, ktoré zachytáva pomocou rozhrania v monitorovacom režime. Získané hodnoty posiela riadiacej jednotke. AC spracuje prijaté informácie a určí, aké mechanizmy sa majú vykonať.
Na monitorovanie stavu siete sa dajú manažmentové rámce, ktoré v sebe nesú informáciu o indexe sile signálu (RSSI), ktorý je prijímaný na prijímači – t.j. bezdrôtovej sieťovej karte WTP zariadenia. Tieto štatistiky sa využívajú pre rozhodovanie o uskutočnení prechodu medzi prístupovými bodmi. Zo všetkých manažmentových rámcov prijatých na monitorovacom rozhraní sa vyberie informácia o sile signálu (RSSI) pre dané STA a posiela sa na AC vo forme štatistického údaju. V správe je STA identifikovaná svojou MAC adresou.
Ak AC zistí pomocou informácií z monitorovania siete, že daná stanica má prejsť na iný WTP začne vykonávať prechod. Najprv oznámi novému WTP zariadeniu, že mobilná stanica sa dostala do jeho dosahu a má tam najlepší signál. Nové WTP, t.j. prístupový bod, na ktorý by sa mala

stanica presunúť, odpovie, či je schopný prijať túto stanicu. V prípade akceptovania mobilnej stanice AC požiada pôvodný prístupový bod, ktorý sa staral o komunikáciu stanice, aby zaslal kontext pre danú stanicu. AC prepošle túto informáciu novému WTP a následne presmeruje dátovú prevádzku na neho. Na základe tejto informácie si nové WTP nastaví požadované parametre a začne komunikovať so stanicou.

V prípade **SDN** sietí presmerovanie dátovej prevádzky pri prechode klienta medzi WTP sa rieši aktualizáciu pravidiel vo *Flow* tabuľkách na **SDN** prepínačoch. AC musí najprv požiadať **SDN** kontrolóra, aby po rozhodovaní vygeneroval príslušné riadiace správy pre prepínače pre efektívne presmerovanie dátového toku patriaceho k danému STA smerom na nové WTP.

Keďže komunikačným médiom je vzduch, stanica sa nestará, kde sa nachádza WTP, cez ktoré momentálne pristupuje do siete, ale komunikuje s ním na základe adresy – BSSID. Roaming je preto úplne transparentný a stanica nijak nepocíti zmenu, pretože komunikuje stále s tou istou cieľovej fyzickou adresou druhej vrstvy.



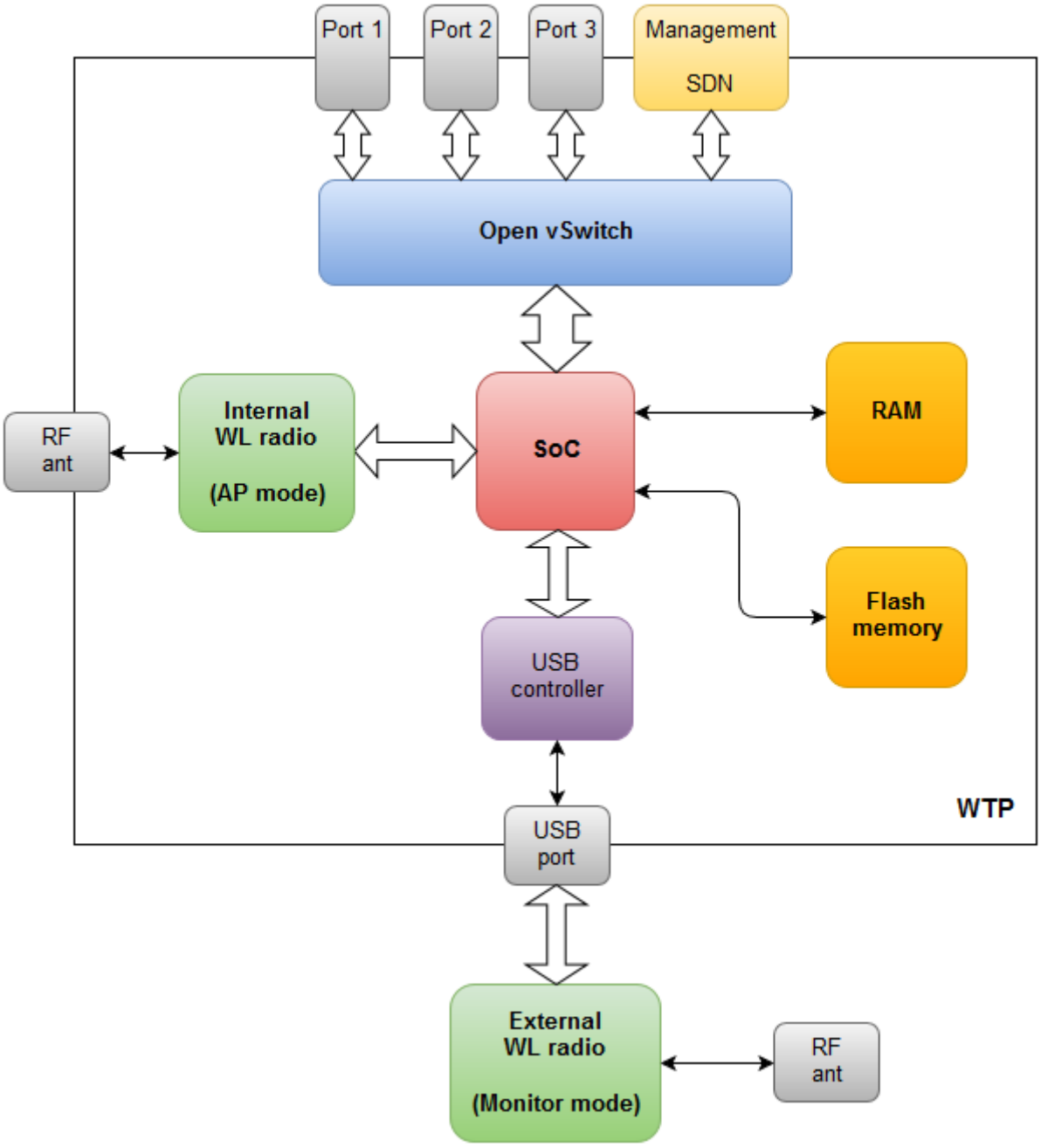
Obr.č.29 - *Sekvenčný diagram roamingu*

3.2.5 Návrh architektúry WTP

Navrhnutá architektúra vnoreného systému pre WTP sa skladá z nasledovných blokoch:

- Systémový čip (SoC) – obsahuje CPU
- Pamäť RAM
- Pamäť Flash vo veľkosti minimálne 8MB
- Programovateľný prepínač s podporou VLAN a Trunk
- Vnútorný bezdrôtový modul s podporou pre virtuálne bezdrôtové rozhrania
- USB modul + aspoň 1 USB port
- Externý bezdrôtový modul pripojený k WTP cez rozhranie USB s podporou monitorovacieho režimu

Architektúra bola navrhnutá tak, aby bola kompatibilná s existujúcim hardvérom *Asus RT-N16* a operačným systémom (firmvérom) *OpenWRT*.



Obr.č.30 - *Bloková schéma architektúry WTP*

3.3 AFCP – Additional Functionality of Control Plane

3.3.1 Riadiaca rovina (Control Plane)

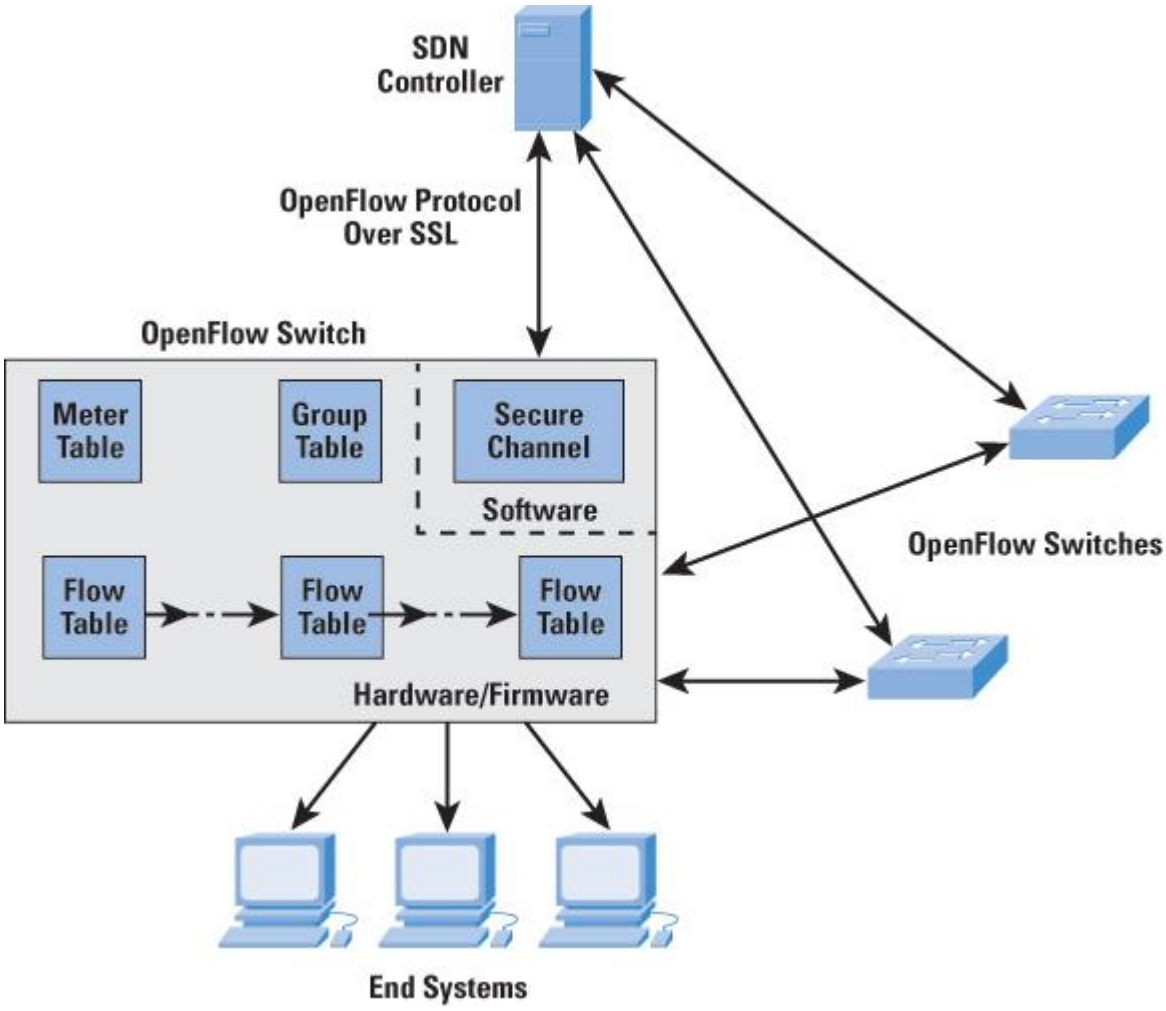
V smerovaní je riadiaca rovina súčasťou architektúry smerovača, ktorá sa zaoberá kreslením topológie siete alebo informáciami v smerovacích tabuľkách, ktoré hovoria, čo robiť s prijatými paketmi. Funkcie riadiacej roviny bežia v architektonickom riadiacom elemente.

Hlavnou funkciou je určiť, ktoré cesty budú v hlavnej smerovacej tabuľke. Multicast smerovanie môže vyžadovať ďalšiu smerovaciu tabuľku pre multicast cesty. Riadiaca rovina zahŕňa funkcie konfigurácie a správy systému.

Obyčajne sa riadiaca rovina nachádza vo firmvéri smerovača spolu s dátovou rovinou, v SDN sieťach je však implementovaná v softvéri. Implementácia riadiacej roviny v softvéri umožňuje dynamický prístup a administráciu siete, poskytuje programový prístup, a teda robí administráciu siete flexibilnejšou. Správca siete tak môže riadiť prevádzku siete z centralizovanej riadiacej konzoly a môže meniť pravidlá prepínačov v sieti (napr. prideliť vysokú prioritu či blokovať niektoré typy paketov).

3.3.2 Flow tabuľky

OpenFlow definuje tri typy tabuliek. Flow tabuľka smeruje pakety na konkrétny tok a špecifikuje funkcie, ktoré majú byť s paketmi vykonané. Flow tabuliek môže byť viac, pričom fungujú tzv. zreťazeným spracovaním. Flow tabuľka môže nasmerovať tok do Group tabuľky, ktorá môže vyvolať rôzne akcie vplývajúce na jeden alebo viacero tokov. Meter tabuľka môže vyvolať rôzne s výkonom súvisiace akcie na tok.



Obr.č.31 – *OpenFlow prepínač* [1]

Komponenty Flow tabuliek

Každý paket vstupujúci do SDN prepínača prechádza cez jednu alebo viaceré Flow tabuľky. Každá Flow tabuľka obsahuje záznamy pozostávajúce zo 6 komponentov:

- *Match fields* – slúži na výber paketov, ktoré zodpovedajú hodnotám v poliach.
- *Priority* – relatívna priorita záznamov tabuľky.
- *Counters* – aktualizované pre zodpovedajúce pakety, sú to rôzne typy časovačov.
- *Instructions* – akcie, ktoré treba vykonať, ak nastane zhoda paketu so záznamom v tabuľke.
- *Timeouts* – maximálny čas nečinnosti predtým ako je tok expirovaný.
- *Cookie* – nepriehľadná hodnota dát vybraná kontrolórom, ktorá môže byť použitá na filtrovanie štatistík tokov, modifikácie tokov a vymazania tokov.

Flow tabuľka môže zahŕňať tzv. „table-miss“ záznam, ktorý má najmenšiu prioritu (0) a každé pole z *Match Fields* má hodnotu „wildcard“, čo znamená, že zodpovedá ľubovoľnej hodnote. Komponent *Match Fields* sa skladá z nasledujúcich polí:

- *Ingress Port* – identifikátor portu prepínača, na ktorý paket prišiel, pričom to môže byť fyzický alebo prepínačom definovaný virtuálny port.
- *Ethernet Source and Destination Addresses* – zdrojová a cieľová MAC adresa.
- *IPv4 or IPv6 Protocol Number* – číslo protokolu indikujúce nasledujúcu hlavičku v pakete.
- *IPv4 or IPv6 Source Address and Destination Address* – zdrojová a cieľová IP adresa.
- *TCP Source and Destination Ports* – zdrojový a cieľový TCP port.
- *User Datagram Protocol (UDP) Source and Destination Ports* - zdrojový a cieľový UDP port.

Predchádzajúce polia sú podporované každým OpenFlow prepínačom. Niektoré OpenFlow prepínače môžu však podporovať aj nasledujúce polia:

- *Physical Port* – používa sa na označenie fyzického portu, keď je paket prijatý na logickom porte.
- *Metadata* – ďalšie informácie, ktoré môžu byť posunuté z jednej tabuľky do inej počas spracovávania paketu.
- *Ethernet Type* – typ Ethernetu.
- *VLAN ID and VLAN User Priority* – polia v IEEE 802.1Q Virtual LAN hlavičke.
- *IPv4 or IPv6 DS and ECN* – polia diferencovaných služieb a explicitných notifikácií zahltenia.
- *Stream Control Transmission Protocol (SCTP) Source and Destination Ports* – zdrojový a cieľový SCTP port.
- *Internet Control Message Protocol (ICMP) Type and Code Fields* – ICMP polia.
- *Address Resolution Protocol (ARP) Opcode*
- *Source and Target IPv4 Addresses in Address Resolution Protocol (ARP) Payload* – zdrojové a cieľové IPv4 adresy v ARP.
- *IPv6 Flow Label*
- *ICMPv6 Type and Code fields* – ICMPv6 polia.
- *IPv6 Neighbor Discovery Target Address* – v IPv6 správe o zistení suseda.
- *IPv6 Neighbor Discovery Source and Target Addresses* – nastavenia adresy v IPv6 správe o zistení suseda na linkovej vrstve.
- *Multiprotocol Label Switching (MPLS) Label Value, Traffic Class, and Bottom of Stack (BoS)* – polia vo vrchnom labeli MPLS label zásobníka.

Komponent *Instructions* pozostáva zo sady inštrukcií vykonávaných, ak paket zodpovedá záznamu. Existujú 2 typy inštrukcií – akcie a sady akcií. Akcie opisujú posielanie paketov, modifikácie paketov a operácie spracúvania Group tabuľky. OpenFlow zahŕňa tieto akcie:

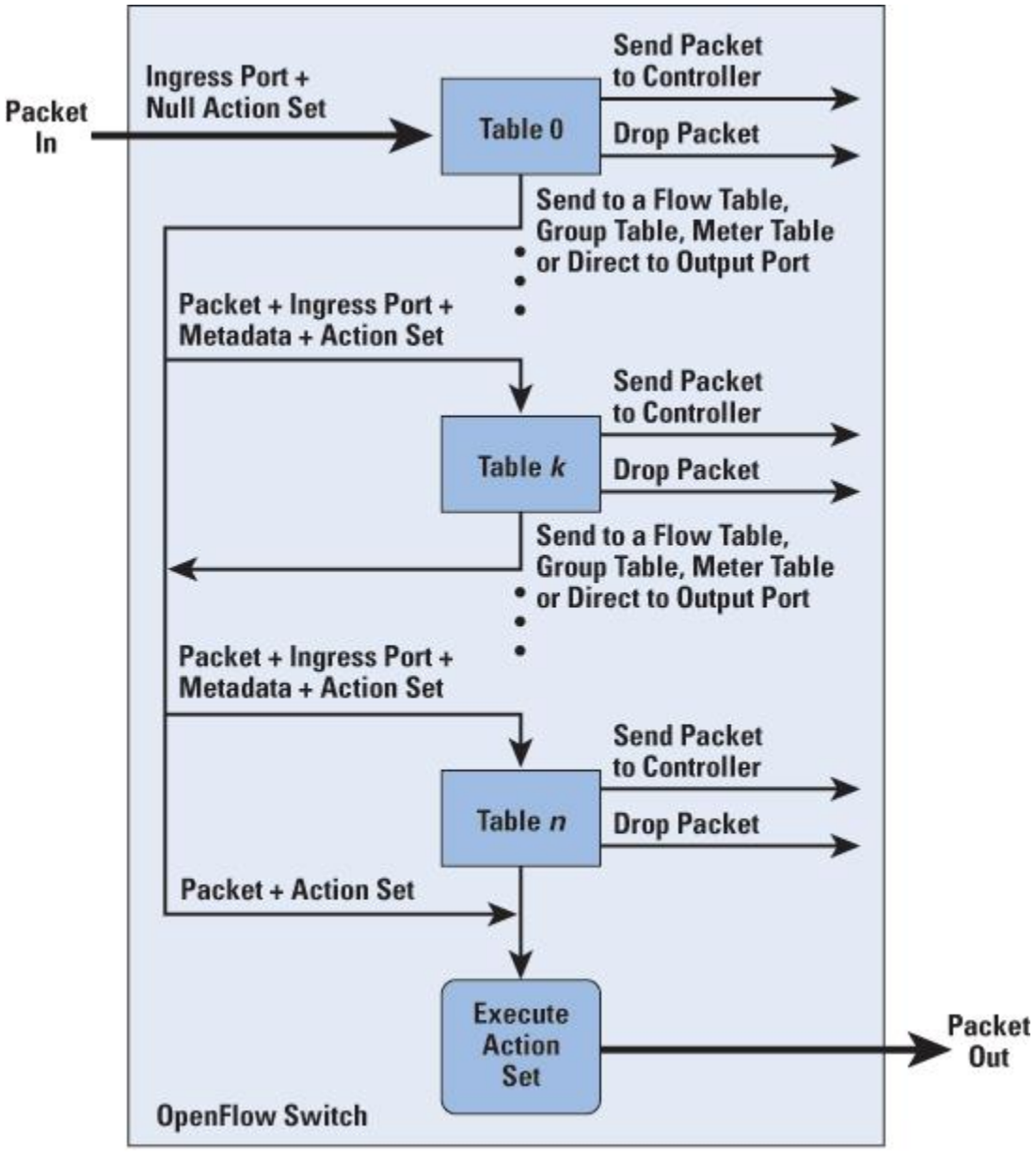
- *Output* – poslanie paketu na konkrétny port.
- *Set-Queue* - nastaví ID fronty pre paket. Keď je paket poslaný na port, ID frontu určuje, ktorý front pripojený k tomuto portu sa použije pre plánovanie a posielanie paketu.
- *Group* – spracovanie paketu cez špecifickú skupinu.
- *Push-Tag/Pop-Tag* – vloží alebo odstráni tag pre VLAN alebo MPLS paket
- *Set-Field* – rôzne akcie, ktoré sú identifikované podľa typu poľa, modifikujú hodnoty príslušných hlavičkových polí paketu.
- *Change-TTL* – rôzne akcie, ktoré modifikujú IPv4 Time To Live (TTL), IPv6 Hop Limit alebo MPLS TTL v pakete.

Sady akcií sú zoznamy akcií spojených s paketom, ktoré sú nahromadené, kým je paket spracovávaný každou tabuľkou a vykonávané, keď paket vystupuje zo zreťazeného spracovania. Existujú štyri typy:

- *Direct packet through pipeline* – inštrukcia *Goto-table* smeruje paket do nasledujúcej tabuľky zreťazeného spracovania. Inštrukcia *Meter* smeruje paket do špecifického merača.
- *Perform action on packet* – akcie s paketom môžu byť vykonávané, keď paket zodpovedá záznamu v tabuľke.
- *Update action set* – zlúčenie akcií do sady akcií pre daný paket v danom toku, alebo zrušenie všetkých akcií v sade akcií.
- *Update metadata* – hodnota metadát môže byť spojená s paketom, čo sa používa na prenos informácií z jednej tabuľky do inej.

Flow tabuľky - zreťazené spracovávanie

Každý SDN prepínač obsahuje jednu alebo viac Flow tabuliek. Ak ich obsahuje viac, sú číslované od 0 a sú zreťazené za sebou.



Obr.č.32 – *Spracovanie paketov cez Flow tabuľky* [1]

Keď je paket predložený tabuľke pre porovnanie so záznamami, vstup pozostáva z paketu, identity vstupného portu, príslušnej hodnoty metadát a príslušnej sady akcií. Pre prvú tabuľku (tabuľka 0) je hodnota metadát prázdna a sada akcií je nulová. Potom proces prebieha nasledovne:

1. Nájdenie zhodného záznamu z najvyššou prioritou vo Flow tabuľke. Ak tu nie je zhoda so žiadnym záznamom a neexistuje „table-miss“ záznam, paket je zahodený. Ak tu je zhoda iba s „table-miss“ záznamom, vykoná sa jedna z troch akcií:
    a. Paket sa pošle na SDN kontrolór, čo povolí kontrolóru definovať nový tok pre tento a jemu podobné pakety, alebo rozhodnúť o zahodení paketu.
    b. Paket sa pošle k nasledujúcej Flow tabuľke v reťazci.

c. Paket sa zahodí.

2. Ak sa nájde zhoda s iným záznamom okrem „table-miss“ záznamu, záznamu sa priradí najvyššia priorita a môžu nasledovať nasledujúce akcie:
   a. Aktualizujú sa počítadlá spojené s týmto záznamom.
   b. Vykonajú sa inštrukcie spojené s týmto záznamom.
   c. Paket je poslaný k nasledujúcej Flow tabuľke v reťazci, na Group tabuľku alebo na Meter tabuľku, alebo môže byť smerovaný na výstupný port.

Keď je paket poslaný na výstupný port, vykoná sa nahromadená sada akcií a paket je vo fronte pre výstup.

OpenFlow protokol

OpenFlow protokol opisuje výmenu správ medzi OpenFlow kontrolórom a OpenFlow prepínačom. Protokol je typicky implementovaný nad SSL alebo TLS (Transfer Layer Security) a poskytuje bezpečný OpenFlow kanál. Povoľuje kontrolóru vykonávať akcie pridania, aktualizovania a odstraňovania záznamov Flow tabuliek. Podporuje tri typy správ:

- *Controller-to-Switch* – správy iniciované kontrolórom a v niektorých prípadoch vyžadujú odpoveď prepínača. Tieto správy umožňujú kontrolóru riadiť logický stav prepínača vrátane konfigurácie a detailov záznamov vo Flow a Group tabuľkách. Medzi tieto správy patrí aj správa „Packet-out“, ktorá sa využíva, keď prepínač posiela paket kontrolóru a ten sa rozhodne nezahodiť paket, ale poslať ho na výstupný port.
- *Asynchronous* – správy odosielané bez zaťažovania kontrolóra. Patria sem rôzne stavové správy posielané kontrolóru. Patrí sem aj správa „Packet-In“, ktorá sa používa, keď prepínač posiela paket kontrolóru, pretože sa nenašla zhoda v záznamoch.
- *Symmetric* – správy odosielané bez obťažovania kontrolóra a prepínača. Sú jednoduché, ale užitočné.

Tabuľka správ:

Správa	Popis
Controller-to-Switch	
Features	Kontrolór žiada o schopnosti prepínača, prepínač pošle odpoveď, ktorá špecifikuje jeho schopnosti.
Configuration	Kontrolór nastavuje a žiada o konfiguračné parametre, prepínač pošle odpoveď s nastaveniami týchto parametrov.

Modify-State	Pridanie, vymazanie a modifikovanie flow/group záznamov a nastavenie vlastností portu na prepínači.
Read-State	Kontrolór zozbiera informácie z prepínača, ako napr. súčasnú konfiguráciu, štatistiky či schopnosti.
Packet-out	Smeruje paket na konkrétny port prepínača.
Barrier	Tieto správy sa používajú kontrolórom pre zabezpečenie, že závislosti správy boli splnené alebo pre prijatie notifikácií pre kompletné operácie.
Role-Request	Nastavenie alebo žiadanie o rolu OpenFlow kanála, čo je užitočné pri pripojení prepínača k viacerým kontrolórom.
Asynchronous-Configuration	Nastavenie filtra pre asynhrónne správy alebo žiadosť o tento filter, čo je užitočné pri pripojení prepínača k viacerým kontrolórom.
Asynchronous	
Packet-In	Transfer paketu na kontrolór.
Flow-Removed	Informovanie kontrolóra o vymazaní záznamu z Flow tabuľky.
Port-Status	Informovanie kontrolóra o zmene portu.
Error	Informovanie kontrolóra o problémovom alebo chybovom stave.
Symmetric	
Hello	Výmena správ medzi kontrolórom a prepínačom pri spustení pripojenia.
Echo	Tieto správy môžu byť posielané buď prepínačom alebo kontrolórom, pričom musí byť prijatá odpoveď. Slúžia na meranie oneskorenia či šírky pásma kontrolór-prepínač pripojenia, alebo iba overujú, že zariadenie je v prevádzke.
Experimenter	Pre ďalšie funkcie, ktoré majú byť súčasťou budúcich verzií OpenFlow.

3.4 HDS - Handover Decision Server

Rozhodovací server prechodov HDS (z angl. Handover Decision Server) – má na starosti funkcionalitu o rozhodovaní prechodov v sieti. Pre uskutočňovanie zmien dátové toku v sieti využíva komponent AFCP, ktorý ma na starosti doplnenie funkcionality riadiaceho plánu.

Inicializácia

HDS odošle správu so svojou konfiguráciou na AFCP. AFCP rozpošle skrze kontrolér broadcastom informáciu WTP, že ak sa takýto protokol spáruje s protokolom vo flow tabuľke má ho posielať na HDS, ak nie tak je potrebné posielať protokol na Kontrolór, ktorý rozhodne čo s ním.

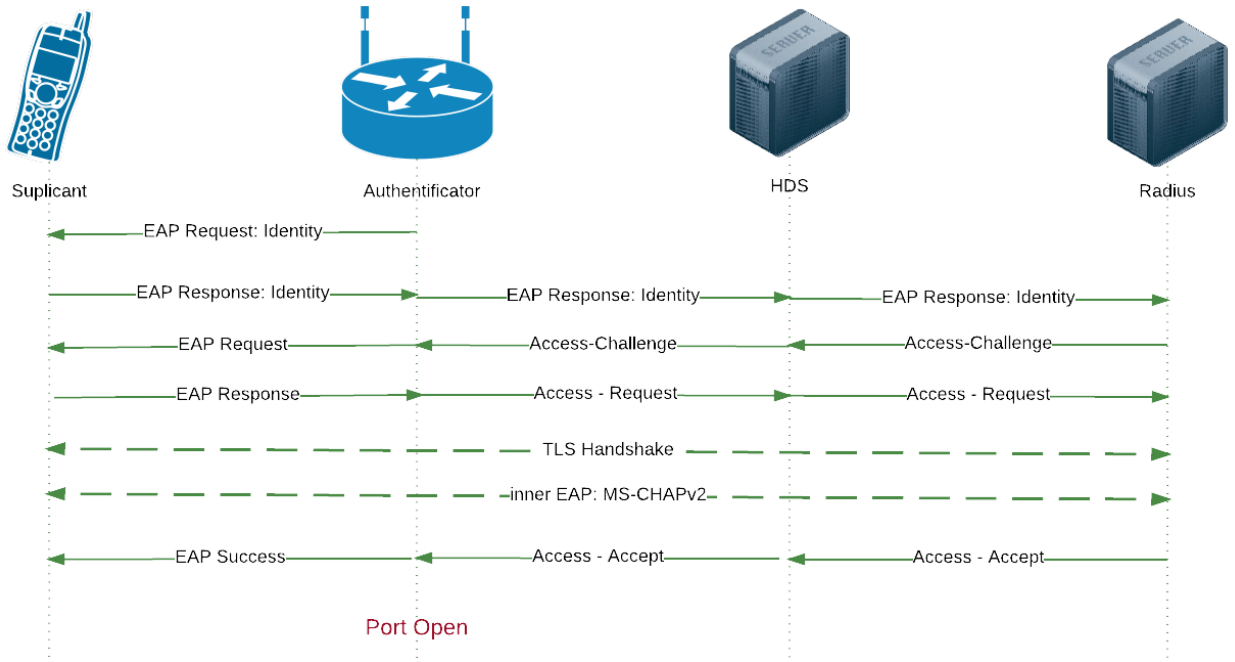
3.4.1 Scenár autentifikácie Suplicant <->HDS <-> Radius

HDS vykonáva komunikáciu s Radius serverom pre identifikáciu a prihlásenie koncového zariadenia. Vykonáva sa tu výmena správ pomocou štandardu 802.1 x

Vstupy a výstupy potrebné pre vykonanie autentifikácie sú opísané na obrázku číslo 33.

Proces: Pokiaľ sa používateľ pripojí na sieťový port, má blokovanú všetkú komunikáciu okrem EAP protokolu, ktorý zaisťuje autentizáciu. Autentizácia prebieha nasledovne:

1. Klient sa pripojí k prístupovému bodu
2. Prístupový bod akceptuje iba autentizačné EAP rámce
3. ostatný (datový) tok od klienta je zablokovaný
4. klient odošle autentizačné informácie pomocou EAP protokolu
5. prístupový bod prepošle žiadosť na HDS server a ten prepošle informácie RADIUS serveru
6. na RADIUS servery prebehne overenie používateľa
7. pokiaľ je používateľ lokálny prebehne jeho overenie priamo na RADIUS servery
8. výsledku autentizácie je informovaný HDS server, ktorý preposiela informácie prístupovému bodu, ktorý v prípade úspechu odblokuje klientovi datový tok.



Obr.č.33 - *Autentifikácia koncového zariadenia*

3.4.2 Scenár prihlásenia bez autentifikácie WTP <-> HDS <-> AFCP

Vstupy: *hw* -probe správy, autentifikačné správy, potvrdzovacie správy

Výstupy: *hw* - správa obsahujúca VAP, BSSID, WTP

hh - riadiace správy na nastavenie toku smerom na HDS (správy:probe, autentifikačné, ACK)

Proces: HDS odošle správu na nastavenie open flow tabuliek do AFCP. Táto správa má za následok preposielanie všetkých probe správ na HDS. Následne je potrebné odoslať správu na AFCP pre nastavenie open flow tabuliek pre autentifikačné správy. Po nastavení open flow tabuliek pre autentifikáciu je potrebné nastavenie open flow tabuliek pre potvrdzovanie autentifikácie taktiež pomocou správy odoslanej na AFCP. Následne môže prebiehať komunikácia pre autentifikáciu.

HDS prijíme od WTP probe správu, ktorú spracuje. Spracovaním sa chápe výber WTP, ktoré je ku zariadeniu najbližšie. Vyberie sa najlepšie WTP na základe prijatej sily signálu. HDS vyberie WTP a odošle mu VAP pre nové koncové zariadenie, ktoré si HDS uloží do tabuľky. HDS následne obrží potvrdzovaciu správu od WTP o vytvorení asociácie. HDS po obdržaní správy odošle správu AFCP. Táto správa bude obsahovať nastavenie toku pre VAP a koncové zariadenie cez konkrétne WTP. Následne už prebieha dátový tok.

3.4.3 Scenár reasociácia WTP <->HDS <-> AFCP

Vstupy: *hw* - SNR, kanál, VAP, informácie o okolitých WTP, MAC (id koncového zariadenia).

Výstupy: *hh* - stará VAP, nová VAP, id koncového zariadenia, MAC, IP, WTP

Proces: Komunikácia slúži na zachytenie a vykonanie zmien v architektúre pri prechode koncovej stanice z jedného WTP na iné WTP. Rozhodovanie záleží, či WTP podporuje správy CSA, alebo nie. Na komusinkáciu medzi HDS a AFCP slúži protokol *hh.* Tento protokol je bližšie opísaný v integračnom manuály.

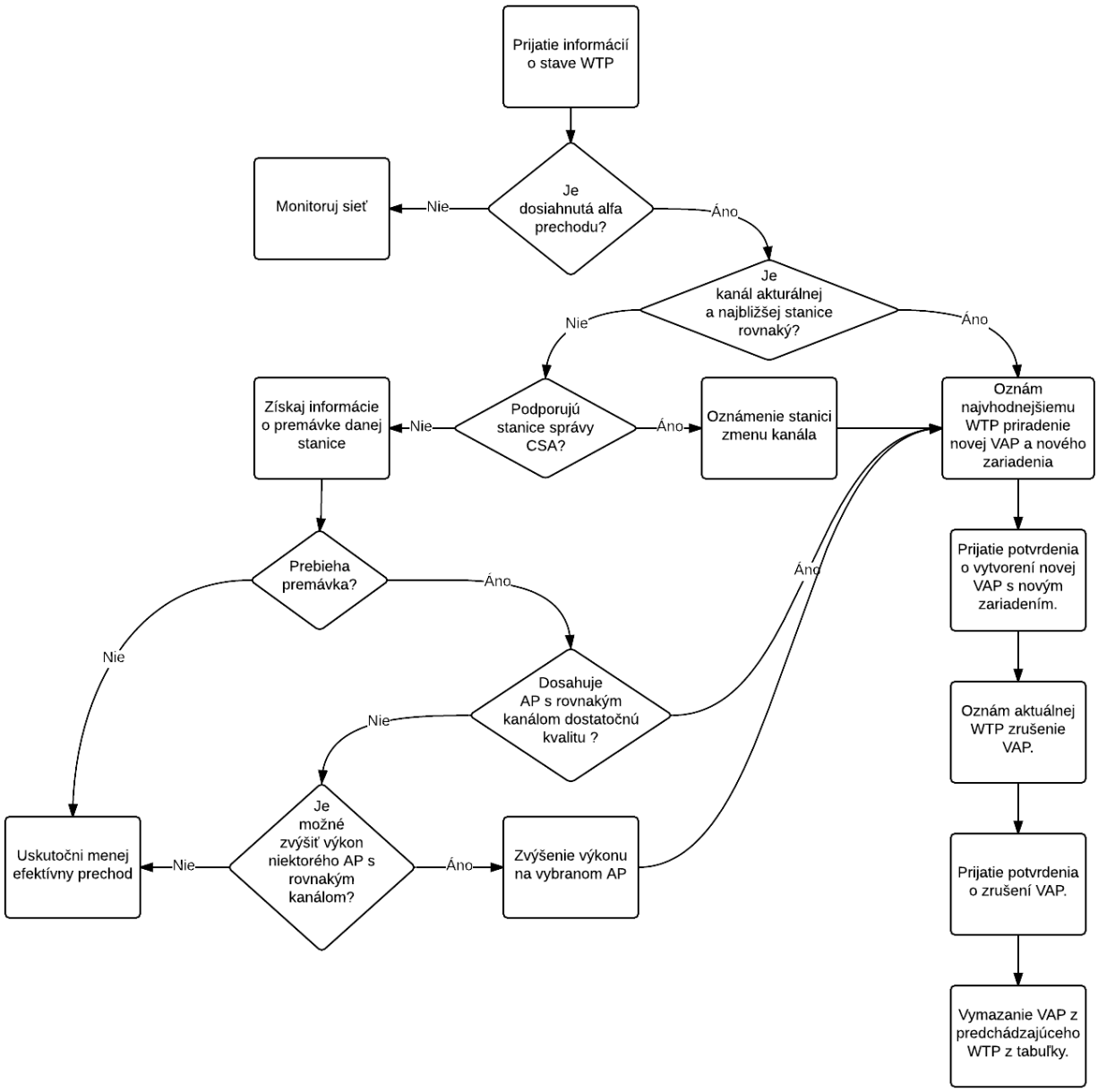
HDS slúži na udržiavanie prehľadu o pripojených zariadeniach v rámci koncových bodov. WTP odosiela informácie ako silu signálu, kanál na ktorom beží a informácie o okolitých staniciach WTP. Z týchto informácií sa vytvorí tabuľka s prehľadom, ktorá bude slúžiť na rozhodovanie pri prechode koncového zariadenia z jednej WTP do inej WTP.

Pre rozhodovanie sa používa HDS , ktorý odosiela WTP informácie o prechode VAP na inú WTP. Následne ako sa vykoná zmena HDS odošle správu AFCP na zmenu toku WTP1 na WTP2, pretože VAP bola premiestnená z WTP1 na WTP2.

Správy z WTP sa budú zasielať každú sekundu.

Postup rozhodovania o uskutočnení prechodu na základe prijatých informácií

Na obrázku číslo 34 je znázornený diagram rozhodovania HDS po prijatí informácií z WTP.



Obr.č.34 - *Rozhodovanie o prechode koncového zariadenia v HDS*

3.4.4 Scenár odhlásenia

Vstupy: *hw* - kanál, VAP, informácie o WTP, MAC (id koncového zariadenia).

Výstupy: odhlásené koncové zariadenie, vymazané z tabuľky v HDS

Proces: Koncové zariadenie odošle na WTP správu pre odhlásenie (definovaná v komunikačnej matici). WTP túto správu prepošle na HDS. HDS prijme správu pomocou rozhrania *hw*. HDS sú pracuje správu, vymaže koncové zariadenie s informáciami z tabuľky a odošle WTP správu o vymazaní.

3.4.5 Scenár monitorovania

Vstupy: *hw* - SNR, kanál, VAP, informácie o okolitých WTP, informácie o WTP (zahŕňa aj obsadenosť WTP), MAC (id koncového zariadenia).

Výstupy: *hh* - zmena kanála, zvýšenie výkonu WTP

Proces: WTP odosiela monitorovacie správy do HDS. HDS prijme správy pomocou rozhrania *hw*. Následne HDS správy spracuje a vykonáva porovnanie s nastavenými pravidlami. V prípade, ak je WTP plne obsadené a chce sa na neho pripojiť ďalšia koncová stanica, je potrebné rozhodnúť, ktorá najbližšia koncová stanica má dostatočné pokrytie, aby dokázala prijať koncové zariadenie a poskytla mu dostatočné QoS. V prípade, ak WTP má rušenie s iným kanálom , je potrebné prestaviť kanál, tak aby nekolidoval s ostatnými prekrývajúcimi sa kanálmi. Všetky správy s rozhodnutiami sa posielajú na AFCP prostredníctvom *hh* rozhrania.

3.4.6 Scenár reportov

Vstupy: *hw* - SNR, kanál, VAP, informácie o okolitých WTP, informácie o WTP (zahŕňa aj obsadenosť WTP), MAC (id koncového zariadenia).

Výstupy: Informácie spracované pre monitorovanie.

Proces: Reporty sú posielané z WTP na HDS pomocou rozhrania *hw*. Tieto reporty sú spraované pri monitorovaní siete, ktoré je opísané v kapitole 5.

3.4.7 Scenár pádu WTP

Vstupy: *hw* - Reporty z WTP

Výstupy: *hh* - správa obsahujúca (staré WTP, nové WTP, VAP, BSSID koncového zariadenia, MAC a IP koncového zariadenia)

hw - správa žiadosti (vyžiadanie reportu)

Proces: HDS prijíme reporty od WTP pomocou rozhrania *hw*. Pri vyhodnocovaní zistí, že WTP nie je funkčné. Odošle správu žiadosti na WTP cez rozhranie *hw*. Ak WTP neodošle v určenom čase 0,5 sekundy žiaden report, HDS odosiela túto správu znovu. Ak ani an druhý pous WTP neodošle správu, HDS vyhodnotí WTP ako nekatívne a odošle všetky informácie na AFCP pomocou rozhrania *hh* na vykonanie prechodu VAP a koncových staníc do iných WTP na základe rozhodnutia. Po vykonaných zmenách HDS očakáva reporty od WTP, kde kontroluje priradenie koncových staníc s VAP do nových WTP. Ak reporty prídu a HDS vyhodnotí, že všetky stanice boli priradené novým WTP pokračuje v monitorovaní siete. Ak ale HDS zistí nepriradenie koncových staníc na nové WTP ani po prijatí druhého reportu, HDS znovu odosiela správu žiadosti do AFCP pre priradenie koncových staníc.

4 Implementácia

4.1 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
```

4.2 Nasadenie firmvéru na smerovač

Táto časť sa venuje problematike smerovačov a OpenFlow. Pre potreby tímového projektu je nutné, aby naše zariadenie podporovalo centralizované ovládanie a dokázalo fungovať ako SDN (softvérovo riadený) prepínač. Bežne takúto funkcionalitu nemá žiadny smerovač, ale po analýze sa nám podarilo zistiť, že je možnosť prerobiť takmer hocijaký smerovač na prepínač s podporou OpenFlow a centralizovaným prístupom. Ako základ slúži niektorý z dvojice open-source firmvérou OpenWrt alebo DD-Wrt. Spočiatku sme sa uberali cestou DD-Wrt, ale po množstve komplikácií a nevyriešených problémov, čiže v dôsledku neúspešnej doimplementácie sme sa rozhodli zmeniť naše orientovanie na OpenWrt. Týmto dokumentom ilustrujeme ako zmeniť bežný komerčný smerovač prepálený s OpenWrt firmvérom na OpenFlow prepínač.

Počas práce bola využitá zatiaľ najnovšia stabilná verzia OpenWrt, ktorá je 15.05 a je známa pod názvom *Chaos Calmer* a je dostupná z webovej stránky OpenWrt. Ako kompatibilný hardvér pre tento firmvér sme mali k dispozícii SOHO smerovač *Asus RT-N16*. Hardvérové parametre tohto smerovača sú:

- CPU: Broadcom BCM4718 SoC 480 MHz (architektúra MIPS 74K)
- Pamäť RAM: 128 MB
- Vnútorná pamäť Flash: 32 MB

- Rozhrania: 4+1 portový gigabitový prepínač, bezdrôtové (WiFi) rozhranie 802.11 b/g/n s max. rýchlosťou 300MB/s, 2x USB 2.0 rozhranie, sériový výstup, JTag
- Napájanie: 12V 1,25A (externý zdroj)

Možnosti doimplementovania OpenFlow do OpenWrt sú hneď dve. Prvá je vziať čistý firmvér OpenWrt vo forme zdrojových kódov, stiahnuť implementáciu čisto protokolu OpenFlow pre OpenWrt, pridať súbory do zdrojových adresárov OpenWrt a potom skompilovať nový firmvér.
Druhou voľbou je pridať softvérový Open vSwitch ako aplikáciu do implementácie OpenWrt.
V tomto dokumente sú popísané návody pre obe vyššie spomenuté možnosti.

4.2.1 Riešenie pomocou OpenFlow implementácie (Stanford implementácia OpenFlow 1.0 a 1.3)

V tejto kapitole sa venujeme doimplementovaniu OpenFlow protokolu do zdrojových kódov. Spočiatku sme sa rozhodli doimplementovať do OpenWrt verziu OpenFlow 1.0. K tejto verzii bola dobrá dokumentácia s názvom projektu Pantou vytvorená univerzitou v americkom Stanforde. Po úspešnom doimplementovaní sme ale zistili, že verzia OpenFlow 1.0 nie je dostačujúca pre potreby nášho tímového projektu a tak sme museli postúpiť na verziu 1.3. Tú sa nám podarilo nájsť samotný modul OpenFlow 1.3 pre OpenWrt bol vytvorený brazílskou nadáciou pre telekomunikácie CPqD. Openflow je implementovaný ako aplikácia na vrchu OpenWrt. Tento návod je rozdelený na tri časti:

- získať prislúchajúci firmvér pre smerovač
  - vložiť firmvér do zariadenia
  - pridanie OpenFlow rozšírenia
- konfigurácia

Pre tento proces je potrebné spĺňať nasledovné požiadavky:

- Operačný systém: Linux distribúcia (otestované s Ubuntu 14.04)
- Internetové pripojenie
- Voľné miesto na disku minimálne 10GB
- Minimálne 1GB dostupnej pamäti RAM

Získanie firmvéru

Rozhodli sme sa vytvoriť firmvér zo zdrojových kódov.
Poznámka: V nasledujúcich krokoch považujeme za pracovný adresár ~/openwrt.
Inštalácia závislostí potrebných pre OpenWrt.

```
apt-get install build-essential binutils flex bison autoconf gettext git \
sharutils subversion libncurses5-dev ncurses-term zlib1g-dev gawk libssl-dev
```

Pre potreby OpenWrt potrebujeme aj program Texinfo, ale v staršej verzii ako je šírený dnes, takže ten musíme nainštalovať ručne.

```
wget http://ftp.gnu.org/gnu/texinfo/texinfo-4.13.tar.gz
gzip -dc < texinfo-4.13.tar.gz | tar -xf -
cd texinfo-4.13
./configure
make
make install
cd ..
```

Stiahneme a pripravíme si zdrojové súbory Chaos Calmer Openwrt.

```
cd ~/openwrt
git clone git://git.openwrt.org/15.05/openwrt.git
cd openwrt
./scripts/feeds update -a
./scripts/feeds install -a
```

Vytvoríme konfiguračný súbor.

```
make menuconfig
```

Tu je dôležité nastaviť target system = Broadcom BCM47xx (MIPS), subtarget = MIPS 74K.
Potom môžeme zatlačiť ESC a potvrdiť uloženie Y.
Skontrolujeme, či máme všetko potrebné pre firmvér.

```
make prereq
```

A napokon spustíme build Chaos Calmer. Prepínač -j2 slúži na využitie viac jadier.

```
make -j2
```

Nahranie firmvéru do zariadenia

Teraz pre overenie je potrebné nahrať firmvér do smerovača. Všetky skompilované firmvéry sú pod zložkou ~/openwrt/bin/brcm. Ten náš sa volá openwrt-brcm47xx-mips74k-asus-rt-n16-squashfs.trx. Overenie sa robí spôsobom:

- Pripojíme kábel do hociktorého "LAN" portu smerovača, nie "WAN"
- PC zmeníme IP adresu 192.168.1.10, maska 255.255.255.0
- Smerovač dáme do recovery režimu - stlačené tlačidlo reset pokým zapájam zdroj
- Recovery režim spoznáme neustálym blikaním symbolu "power" na smerovači
- Akýmkoľvek TFTP klientom pošleme firmvér na IP adresu 192.168.1.1
- Počkáme dve minúty, potom power resetneme smerovač
- Znova počkáme 2 minúty a vyskúšame telnet na 192.168.1.1
- Mala by nás uvítať OpenWrt úvodná obrazovka.

Pridanie OpenFlow rozšírenia

Presunieme sa do pracovného adresára a stiahneme OpenFlow rozšírenie.

```
cd ~/openwrt/
git clone https://github.com/CPqD/openflow-openwrt.git
```

Pridáme symbolickú linku na OpenFlow.

```
cd ~/openwrt/package/
ln -s ~/openwrt/openflow-openwrt/openflow-1.3/
```

Pridáme základné konfiguračné súbory.

```
cd ~/openwrt/
ln -s ~/openwrt/openflow-openwrt/openflow-1.3/files
```

Znova vytvoríme konfiguračný súbor, tento krát už s OpenFlow.

```
make menuconfig
```

Zvolíme nasledovné:

Target system = Broadcom BRCM47xx (MIPS)

Subtarget = MIPS 74k

- Pod network zvolíme balík tc, aby sa nainštaloval
- Pod Kernel Modules -> Network Support zvolíme kmod-tun na inštaláciu
- ukončíme a uložíme

Pridáme podporu pre frontu.

```
make kernel_menuconfig
```

Pod Networking Support -> Networking options -> QoS zvolíme Hierarchical Token Bucket (HTB) na inštaláciu. Ukončíme a uložíme.

Spustíme build.

```
make
```

Nainštalujeme firmvér do smerovača, napríklad vyššie spomenutým spôsobom cez recovery.

Overenie

Základne, firmvér vytvorený zo zdrojových súborov bude mať port označený ako "internet" (WAN) nastavený ako manažovací port, so statickou IP 192.168.1.1. Mali by sme byť schopný pripojiť sa cez tento port ak máme IP adresu PC v podsieti 192.168.1.0/24. Keď sme nakonfigurovali PC, môžeme sa pokúsiť pripojiť.

```
telnet 192.168.1.1
```

Po pripojení overím, či bežia potrebné OpenFlow procesy.

```
ps aux | grep ofprotocol
ps aux | grep ofdatapath
```

Konfigurácia

Pre OpenFlow sa nachádzajú v smerovači tri potrebné konfiguračné súbory. Pre sieť (/etc/config/network) a wifi (/etc/config/wireless) a konfiguráciu (/etc/config/openflow).

Primárne je wifi vypnuté. Tento smerovač vie byť použitý ako 5 portový prepínač. Najskôr musíme nastaviť /etc/config/network.

```
config 'switch'
```

```
        option 'name' 'eth0'
        option 'reset' '1'
        option 'enable_vlan' '1'

config 'switch_vlan'
        option 'device' 'eth0'
        option 'vlan' '1'
        option 'ports' '4 8t'

config 'switch_vlan'
        option 'device' 'eth0'
        option 'vlan' '2'
        option 'ports' '3 8t'

config 'switch_vlan'
        option 'device' 'eth0'
        option 'vlan' '3'
        option 'ports' '2 8t'

config 'switch_vlan'
        option 'device' 'eth0'
        option 'vlan' '4'
        option 'ports' '1 8t'

config 'switch_vlan'
        option 'device' 'eth0'
        option 'vlan' '0'
        option 'ports' '0 8t'

config 'interface' 'loopback'
        option 'ifname' 'lo'
        option 'proto'  'static'
        option 'ipaddr' '127.0.0.1'
        option 'netmask' '255.0.0.0'

config 'interface'
        option 'ifname' 'eth0.1'
        option 'proto' 'static'

config 'interface'
        option 'ifname' 'eth0.2'
        option 'proto' 'static'

config 'interface'
        option 'ifname' 'eth0.3'
        option 'proto' 'static'

config 'interface'
        option 'ifname' 'eth0.4'
        option 'proto' 'static'

config 'interface'
```

```
        option 'ifname' 'eth0.0'
        option 'proto' 'static'
        option type 'bridge'
        option 'ipaddr' '192.168.1.1'
        option 'netmask' '255.255.255.0'
```

Manažovací port je WAN. Tu bude pripojený aj kontrolór. Zvyšné porty LAN1-4 sú použiteľné pre prepínač.

Ešte potrebujeme nastaviť wifi, ktoré je stále vypnuté. Základne je povolené maximálne 802-11g, ale smerovač je schopný fungovať aj v štandarde n. Pre využitie sú tieto príkazy.

```
opkg remove kmod-b43 kmod-b43legacy
opkg update
opkg install kmod-brcmsmac
rm -f /etc/config/wireless
wifi detect > /etc/config/wireless
wifi
```

Nastavíme konfiguračný súbor pre wifi /etc/config/wireless.

```
config wifi-device  wlan0
        option type     mac80211
        option channel  5
        option macaddr   00:25:9c:30:2c:f4
        option hwmode    11n

        # REMOVE THIS LINE TO ENABLE WIFI:
        # option disabled 0

config wifi-iface
        option device   wlan0
#       option network  lan
        option mode     ap
        option ssid     OpenFlow-OpenWrt
        option encryption none
```

Ďalej potrebujeme nastaviť /etc/config/openflow.

```
config 'ofswitch'
        option 'dp' 'dp0'
        option 'ofports' 'eth0.0 eth0.1 eth0.2 eth0.3 eth0.4 wlan0 '
        option 'ofctl' 'tcp:192.168.1.10:6633'  #ip adresa controllera
        option 'mode'  'outofband'
```

Nakoniec spravíme reštart, nech sa všetky zmeny prejavia. Ale pre istotu aj fyzický reštart.

```
/etc/init.d/openflow restart
/etc/init.d/network restart
```

4.2.2 Riešenie pomocou Open vSwitch - Pridanie modulu

Na pridanie balíka Open vSwitch do firmvéru máme k dispozícii 2 možnosti:

1. Pridanie OVS do zdrojového kódu a kompilácia firmvéru
2. Inštalácia hotového balíka OVS pomocou manažéra balíkov na bežiacom systéme

Počas riešenia boli otestované obe metódy. Postup je podrobne opísaný v nasledujúcich častiach tejto podkapitoly.

4.2.2.1 Metóda 1: Pridanie OVS do zdrojového kódu a kompilácia

Pre tento proces je potrebné spĺňať nasledovné požiadavky:

- Operačný systém: Linux distribúcia (otestované s Ubuntu 14.04)
- Internetové pripojenie
- Voľné miesto na disku minimálne 10GB
- Minimálne 1GB dostupnej pamäti RAM

V prvom kroku je potrebné skontrolovať, či sú nainštalované všetky potrebné balíky pre kompiláciu použitím tohto príkazu:

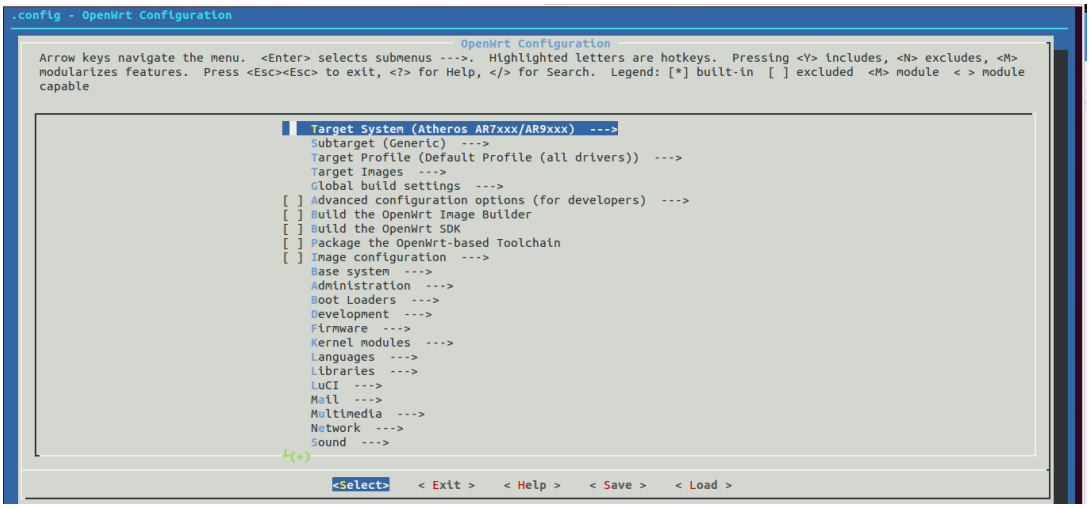
```
$ apt-get install build-essential binutils flex bison autoconf gettext texinfo sharutils
subversion git libncurses5-dev ncurses-term zlib1g-dev gawk
```

Teraz máme na možnosť vybrať si verziu OpenWrt, ktoru chceme použiť. V tomto prípade to bude verzia *Chaos Calmer*. Najnovšia, ale nestabilná sa nazýva vždy ako *trunk.*

Predpokladáme, že sa nachádzame v domovskom adresári, kde vykonáme tieto príkazy:

```
$ svn co svn://svn.openwrt.org/openwrt/branches/chaos_calmer
$ cd chaos_calmer
$ ./scripts/feeds update –a
$ ./scripts/feeds install –a
$ make menuconfig
```

Po vykonaní posledného príkazu by sa malo objaviť automaticky okno s konfiguráciou, ktoré vyzerá podobne ako na obrázku:



Obr.č.35 - *Konfiguračné okno „menuconfig“*

Ako prvé vyberieme si položku *Target system* pomocou <Select>. Následne si vyberieme *Broadcom BCM47xx/53xx (MIPS)* v prípade Asus RT-N16. Pri stlačení kláves je odporúčané sa riadiť pokynmi, ktoré sa vždy objavujú na obrazovke v hornej časti okna *menuconfig*.

Ako *Subtarget* si vyberieme možnosť *MIPS 74K*. (samozrejme v prípade RT-N16)

V časti *Target Profile* máme možnosť vybrať si ovládač pre WiFi. Dôležité je že predvolený ovládač *b43* podporuje maximálne režim 802.11g.

V časti *Kernel modules* a *Network support* potrebujeme ešte *kmod-tun*.

V časti *Network* potrebujeme ešte modul *openvswitch*.

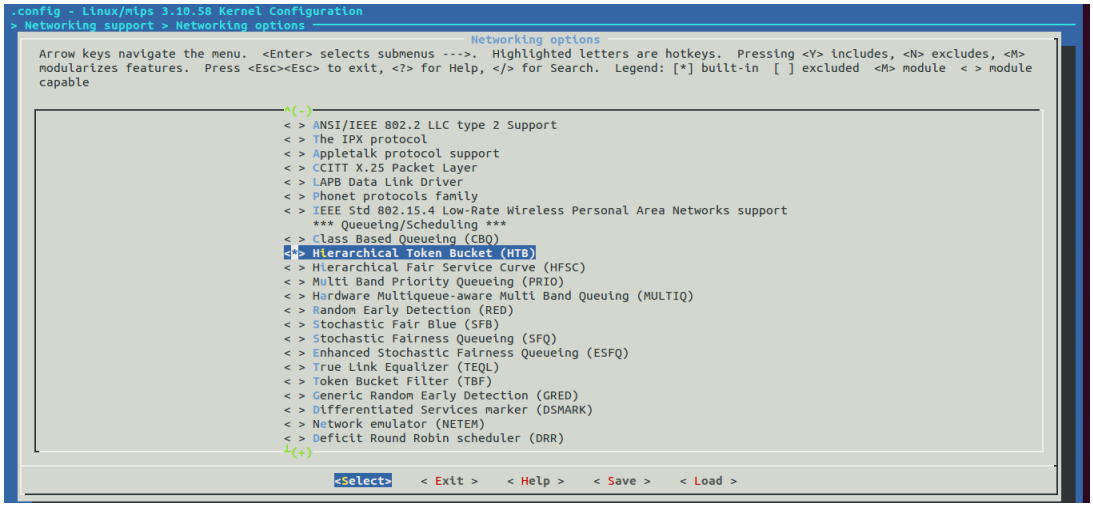
V prípade potreby používateľského rozhrania, máme na možnosť vybrať si a prispôsobiť balík Luci, napríklad takto:

- v časti *Luci -> Collections* si vyberieme *luci*.
- v časti *Luci -> Modules* si vyberieme *luci-base, luci-mod-admin-full*.
- v časti *Luci -> Applications* si vyberieme *luci-app-firewall, luci-app-ntpc*.
- v časti *Luci -> Themes* si zvolíme vzhľad webového rozhrania. Predvolený je *luci-theme-bootstrap*.
- v časti *Luci -> Protocols* si vyberieme *ipv6* a *ppp*.
- v časti *Luci -> Libraries* si vyberieme *httpclient, ip, json* a *nixio*.

Hotovú konfiguráciu si uložíme pomocou tlačidla <Exit> a následne Yes v dialógovom okne. Po návrate do konzoly spustíme príkaz:

```
$ make kernel_menuconfig
```

Objaví sa podobné okno ako *menuconfig* ale tento krát s inými položkami. Na ceste *Networking Support -> Networking Options* si pridáme balík *Hierarchical Token Bucket (HTB)* do kompilácie.



Obr.č.36 - *Konfiguračné okno „Kernel menuconfig“ a položka HTB*

Nasleduje posledný krok tejto metódy a to je spustenie kompilácie, ktorá sa spustí vydaním príkazu:

```
$ make
```

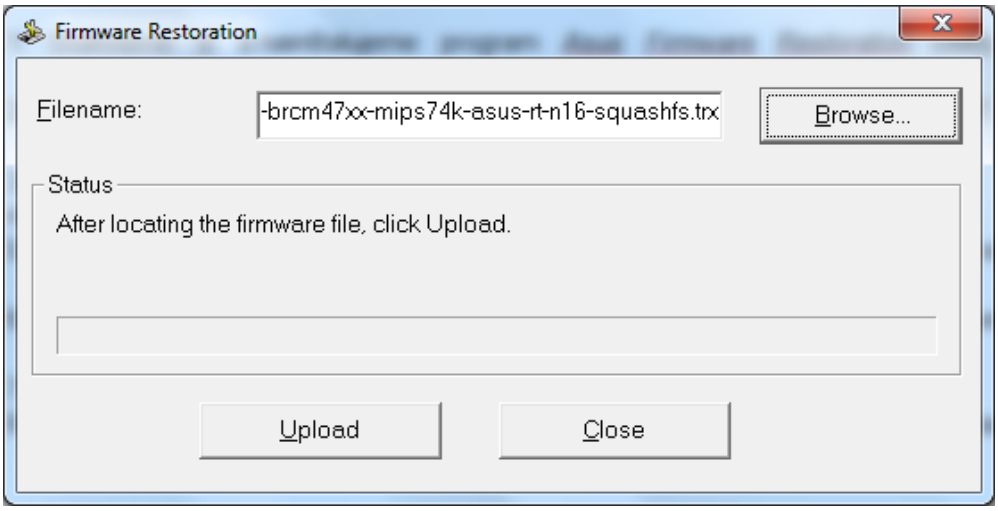
Proces kompilácie môže trvať aj niekoľko hodín. Pre zrýchlenie procesu sa dá zapnúť *multithreading* a to tak že pomocou prepínača *-j* pre *make* pridáme do procesu vykonanie viac úloh naraz. Napríklad v prípade *-j2* sa budú vykonávať dve úlohy naraz, čo je odporúčané množstvo v prípade dvojjadrového procesora a 1GB RAM.

Hotový binárny súbor sa bude nachádzať v adresári *./bin/<SoC_type>* . V prípade RT-N16 to bude: *openwrt-brcm47xx-mips74k-asus-rt-n16-squashfs.trx*.

Inštalácia firmvéru na router RT-N16

Predpokladáme, že máme k dispozícii binárny súbor s príponou *.trx*, ktorý je určený pre dané zariadenie. V nasledujúcich krokoch je opísaný postup inštalácie pri použití operačného systému Windows. Na OS Linux je to možné tiež, napr. pomocou nástroja *tftp*.

1. Stiahneme si a nainštalujeme program *Asus Firmware Restoration Utility* zo stránkach výrobcu.
2. Smerovača si prepneme do tzv. *Recovery* režimu pomocou tlačidla RESET. Po úspešnom prepnutí do tohto režimu sa bude LED PWR neprerušene blikať.
3. IP adresu počítača si nastavíme na *192.168.1.10* a masku na *255.255.255.0*.
4. Pripojíme si zariadenie k počítaču použitím niektorého portu LAN. (napr. LAN1)
5. Spustíme program *Firmware Restoration* a pomocou *Browse* si vyberieme súbor s firmvérom (prípona *.trx*).
6. Klikneme na *Upload* a čakáme kým sa neobjaví informácia o úspešnom dokončení procesu inštalácie. (obrázok číslo 37)
7. Zariadenie sa po inštalácie firmvéru reštartuje.
8. úspešnosti inštalácie sa presvedčíme pomocou nástroja *telnet*. Pripojíme sa na adresu *192.168.1.1*. Mala by sa objaviť konzola s nadpisom OpenWrt.



Obr.č.37 - Okno aplikácie Firmware Restoration

4.2.2.2 Metóda 2: Inštalácia hotového balíka OVS pomocou manažéra balíkov

Táto metóda je jednoduchšia a rýchlejšia ako predchádzajúca, keďže je to bez dlhotrvajúcej kompilácie firmvéru. Nepotrebujeme tu ani prostredie pre kompiláciu. Nevýhodou však je že inštalácia na zariadení trvá dlhšie, keďže potrebné balíky sú nainštalované manuálne pomocou manažéra balíkov *opkg*. Taktiež je nevyhnutná dostupnosť internetového pripojenia na smerovači cez WAN port.

V prvom kroku si stiahneme už kompilovaný firmvér zo stránky OpenWrt pre príslušné zariadenie. V našom prípade je to verzia *Chaos Calmer (15.05)* pre router *Asus RT-N16*. Názov súboru s firmvérom vyzerá nasledovne:

```
openwrt-15.05-brcm47xx-mips74k-asus-rt-n16-squashfs.trx
```

V ďalšom kroku si nainštalujeme tento stiahnutý firmvér na koncové zariadenie podľa návodu vyššie *Inštalácia firmvéru na router RT-N16* (v prípade Asus). Po úspešnej inštalácii by sme mali mať k dispozícii prístup do zariadenia cez *telnet*.

Pomocou príkazu *passwd* je potrebné nastaviť si heslo. Po úspešnom nastavení hesla budeme mať k dispozícii aj prístup cez SSH a cez webového rozhrania *Luci*. Používateľské meno bude vždy „root“.

Potrebné balíky si nainštalujeme do vnútornej pamäti zariadenia pomocou nasledovných príkazov:

```
$ opkg update   # aktualizácia databázy o dostupných balíkoch
$ opkg --force-depends install kmod-tun openvswitch
```

V prípade potreby používateľského rozhrania, máme tu možnosť nainštalovať si a prispôsobiť balík Luci. Minimálna konfigurácia sa inštaluje takto:

```
$ opkg --force-depends install luci
```

Inštaláciu dokončíme reštartovaním zariadenia a pokračujeme s konfiguráciou SDN prepínača.

Konfigurácia prepínača

Predpokladáme, že už máme pripravené zariadenie s firmvérom *OpenWrt*, ktorý obsahuje funkčné používateľské prostredie *Open vSwitch*. Musia bežať procesy *ovsdb-server* a *ovs-vswitchd*.

Nasledovné kroky konfigurácie a obsahy konfiguračných súborov sú kompatibilné predovšetkým s verziou OpenWrt 15.05, Open vSwitch 2.3.9 a zariadením Asus RT-N16. V prípade iných softvérových verzií alebo iného hardvéru obsah niektorých konfiguračných súborov môže vyzerať inak. Postup:

1. Pripojíme sa na IP adresu prepínača (predvolene 192.168.1.1) cez protokol SSH a zadáme používateľské meno (root) a heslo.

2. Pomocou obľúbeného textového editora (napríklad *vi* alebo *nano*) zmeníme obsah niektorých konfiguračných súborov.
   a. Súbor /etc/config/network :

```
config switch
        option name 'eth0'
        option reset '1'
        option enable_vlan '1'

config switch_vlan
        option device 'eth0'
        option vlan '0'
        option ports '1 8t'

config switch_vlan
        option device 'eth0'
        option vlan '1'
        option ports '4 8t'

config switch_vlan
        option device 'eth0'
        option vlan '2'
        option ports '3 8t'

config switch_vlan
        option device 'eth0'
        option vlan '3'
        option ports '2 8t'

config interface 'loopback'
        option ifname 'lo'
        option proto 'static'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'

config globals 'globals'
        option ula_prefix 'fd1a:8ff4:8d69::/48'

config interface 'lan'
        option ifname 'eth0.0'
        option force_link '1'
        option type 'bridge'
        option proto 'static'
        option ipaddr '192.168.1.1'    # zmenit podla potreby
        option netmask '255.255.255.0'
        option ip6assign '60'

config interface
        option ifname  'eth0.1'
        option proto 'static'
```

```
config interface
        option ifname  'eth0.2'
        option proto 'static'

config interface
        option ifname  'eth0.3'
        option proto 'static'
```

Kontrolované porty prepínača budú LAN1-3 a port pre pripojenie kontrolóra bude LAN4. Port WAN je vypnutý.

b. Súbor */etc/config/wireless* :

Odstránime riadky „*option disabled 1“, „option network lan“* a pridáme alebo zmeníme tieto riadky v časti *config wifi-iface*:

```
option ssid 'inWifi'
option encryption 'psk2'    ← podľa potreby
option key 'my_password'    ← podľa potreby, je to heslo typu WPA2-PSK
```

c. Pre správnu funkčnosť WLAN LED je potrebné rozšíriť obsah súboru */etc/config/system* pridaním nasledovných riadkov:

```
config led wlan_led
        option name 'WLAN'
        option sysfs 'bcm47xx:blue:wlan'
        option trigger 'netdev'
        option dev 'wlan0'
        option mode 'link tx rx'
```

3. Po dokončení zmien v konfiguračných súboroch je potrebné reštartovať prepínač.
4. Počítač kde robíme konfiguráciu už musí byť pripojený výhradne do portu LAN4.
5. Spojíme sa so zariadením cez SSH, rovnako ako v predchádzajúcich krokoch.
6. Vytvoríme si most (bridge) pomocou príkazu:

```
$ ovs-vsctl add-br br0
```

Parameter *br0* je názov vytvoreného rozhrania.

7. Do vytvoreného virtuálneho rozhrania *br0* pridáme rozhrania portov. Predpokladáme, že sme využili obsah súboru *network* opísaného vyššie.

```
$ ovs-vsctl add-port br0 eth0.1
$ ovs-vsctl add-port br0 eth0.2
$ ovs-vsctl add-port br0 eth0.3
$ ovs-vsctl add-port br0 wlan0    # ak chceme pridat do switchu aj WiFi
```

8. Nastavíme si IP adresu a port kontrolóra. V našom prípade to bude *192.168.1.10*:6633.

```
$ ovs-vsctl set-controller br0 tcp:192.168.1.10:6633
```

9. Nastavíme si požadovanú/é verziu/e OpenFlow. Prvým príkazom sa povolí iba OpenFlow 1.3, druhým sa povolia verzie 1.0 a 1.3.

```
$ ovs-vsctl set bridge br0 protocols=OpenFlow13
$ ovs-vsctl set bridge br0 protocols=OpenFlow10,OpenFlow13
```

10. Konfiguráciu si overíme pomocou príkazu:

```
$ ovs-vsctl show
```

4.3 Nasadenie a spojazdnenie SDN kontrolóra a prepínača

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
```

Následne môžeme RYU spustiť a začať komunikáciu. Spustíme cvičný OpenFlow 1.3 skript.

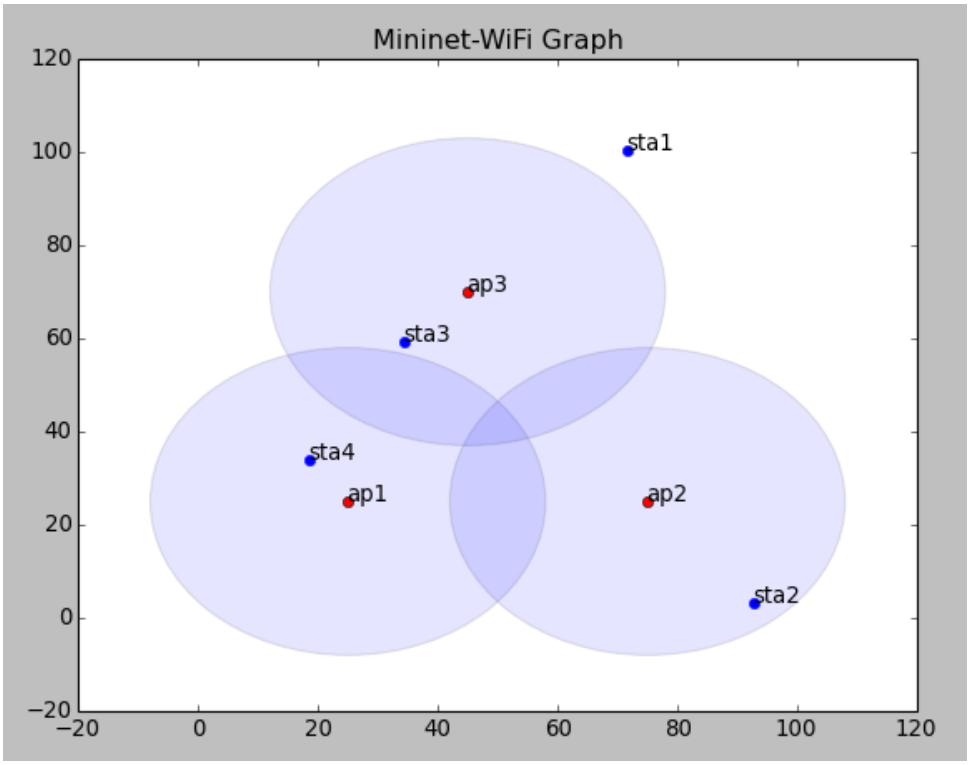
```
% cd bin
% ./ryu-manager ryu/app/simple_switch13.py
```

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.

5 Testovanie

5.1.1 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.38 - *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.

5.2 Simulácia v OpenNet

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.

5.3 Flow tabuľky

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.

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
```

5.4 Vizualizácia topológie

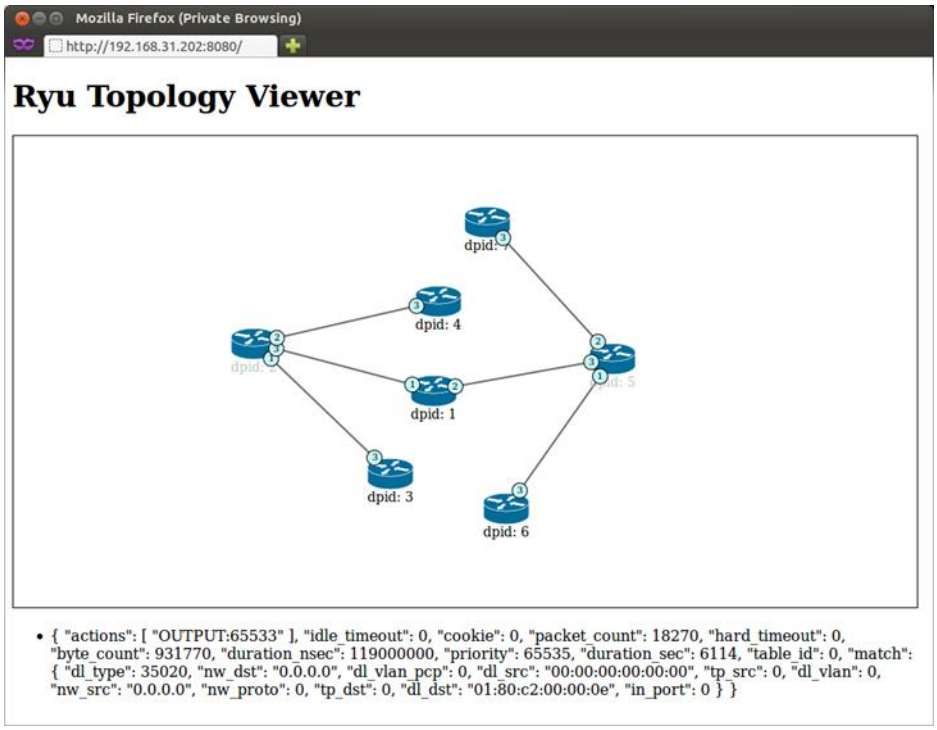
Keď máme našu SDN architektúru korektne zapojenú, môžeme využiť vlažnosti RYU kontrolóra na vizualizáciu našej topológie. Táto vizualizácia závisí od troch aplikácií:

- 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 39.



Obr.č.39 – *GUI na vizualizáciu topológie pomocou RYU* [5]

6 Literatúra

[1] Michal Rjaško: Bezpečnosť bezdrôtových sietí [online], Univerzita komenského, Fakulta matematiky, fyziky a informatiky UK, 2013, [cit: 2013-04-11]. Dostupné na internete: <http://new.dcs.fmph.uniba.sk/files/biti/wifi.pdf>

[2] 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/>.

[3] 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/>.

[4] 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>.

[5] 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>.